



Rejeu de chemin et localisation monoculaire : Application du Visual SLAM sur carte peu dense en environnement extérieur contraint.

Féraud Thomas

► To cite this version:

Féraud Thomas. Rejeu de chemin et localisation monoculaire : Application du Visual SLAM sur carte peu dense en environnement extérieur contraint.. Automatique / Robotique. Université Blaise Pascal - Clermont-Ferrand II, 2011. Français. NNT: . tel-00696256

HAL Id: tel-00696256

<https://theses.hal.science/tel-00696256>

Submitted on 11 May 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : D. U : 2191
E D S P I C : 544



UNIVERSITÉ BLAISE PASCAL - CLERMONT II

École Doctorale SPI

Sciences Pour l'Ingénieur de Clermont-Ferrand

Thèse présentée et soutenue par

Thomas FÉRAUD

en vue de l'obtention du grade de

DOCTEUR D'UNIVERSITÉ

Spécialité : VISION POUR LA ROBOTIQUE

**Rejeu de chemin et localisation monoculaire :
Application du Visual SLAM sur carte peu dense en
environnement extérieur contraint.**

Thèse dirigée par Roland CHAPUIS

préparée au LASMEA

soutenue publiquement le 09 Décembre 2011

Jury :

<i>Rapporteurs :</i>	Udo FRESE	- Assistant professor University of Bremen
	David FILLIAT	- Enseignant-Chercheur UEI-ENSTA
	François CHARPILLET	- Directeur de recherche INRIA/LORIA
<i>Examineurs :</i>	Michel DEVY	- Directeur de recherche CNRS/LAAS
	Romuald AUFRÈRE	- Maître de conférences UBP
	Paul CHECCHIN	- Maître de conférences UBP
<i>Invité :</i>	Zhengcheng HU	- Professeur associé Kumamoto University
	Joël MORILLON	- Thalès Optronique S.A.
<i>Directeur :</i>	Roland CHAPUIS	- Professeur UBP

Remerciements

Tout d'abord, je remercie Roland Chapuis, Paul Checchin et Romulad Aufrère pour m'avoir épaulé durant cette belle aventure qu'est la thèse, pour leurs conseils avisés et leur soutien. Ces trois années de recherches n'auraient pas été les mêmes sans le questionnement et les challenges qui m'ont permis d'apporter un simple grain de sable à l'ouvrage scientifique.

Je remercie également Thalès Optronique et le laboratoire LAAS pour leur participation au sein du projet RINAVEC sans lequel mes travaux de thèses n'auraient pu voir le jour.

Je remercie le LASMEA et plus particulièrement l'équipe Gravir de m'avoir accueilli. Je remercie de même Laurent Malaterre, Serge Alizon pour toute l'aide qu'ils m'ont apporté dans la réalisation des différentes expérimentations que j'ai eu à mener.

Je remercie aussi l'équipe du professeur Zhencheng Hu de l'Université de Kumamoto pour m'avoir accueilli pendant un mois au sein de leur locaux.

Je remercie particulièrement Damien Vivet, Guillaume Bresson, Maxime Pradelle, Florent Malartre, Pierre Delmas, Pierre Guerin et tout ceux qui ont participé à la bonne humeur ou à l'avancée de mes travaux.

Avant de conclure, je remercie tout ceux que je n'ai pas cités et qui se reconnaitront pour être passés dans le bureau 3010, que ce soit pour le travail, pour partager une idée ou pour simplement dire bonjour.

Enfin, je tiens à remercier toute ma famille pour son soutien constant tout au long de mes études et de mon doctorat même si tout n'était pas toujours très évident. De la même manière, je remercie mes amis pour ne m'avoir pas trop poser de questions et pour avoir acquiescé de la plus simple manière qui soit tandis que je m'égarais dans des explications sensées allures autant farfelues que justifiées.

Résumé

Dans le cadre de la robotique mobile en environnement extérieur, les concepts de localisation et de perception sont au coeur de toute réalisation. Aussi, les travaux menés au sein de cette thèse visent à rendre plus robustes des processus de localisation existants sans pour autant augmenter de manière notable leur complexité. La problématique proposée place un robot au sein d'un environnement potentiellement dangereux avec pour objectif de suivre une trajectoire établie comme sécurisée avec une carte aussi simple que possible. De plus, des contraintes fortes sont imposées tant dans la réalisation (système peu onéreux, indétectable) que dans le résultat (une exécution temps-réel et une localisation en permanence dans une tolérance de 10 cm autour de la trajectoire de référence). Le capteur extéroceptif choisi pour mener à bien ce projet est une caméra tandis que l'estimation de la pose du véhicule à chaque instant est réalisée par un filtre de Kalman dans sa version étendue. Les principaux problèmes d'estimation résident dans la non-linéarité des modèles d'observation et les contributions apportées apportent quelques solutions :

- une méthode de calcul exacte de la propagation des incertitudes de l'espace monde vers l'espace capteur (caméra) ;
- une méthode de détection des principaux cas de divergence du filtre de Kalman dans le calcul de la phase de mise à jour ;
- une méthode de correction du gain de Kalman.

Ce projet avait deux objectifs : réaliser une fonction de localisation répondant aux contraintes fortes préalablement évoquées, et permettre à un véhicule de quitter temporairement la trajectoire de référence, suite à la prise en main de l'opérateur pour ensuite reprendre le cours normale de sa mission au plus près de la trajectoire de référence. Ce deuxième volet fait intervenir un cadre plus large dans lequel il faut, en plus de la localisation, cartographier son environnement. Cette problématique, identifiée par l'acronyme SLAM (Simultaneous Localization And Mapping), fait le lien avec les deux dernières contributions de ces travaux de thèse :

- une méthode d'initialisation des points qui constitueront la carte SLAM ;
- une méthode pour maintenir la cohérence entre la carte de référence et la carte SLAM.

Des résultats sur des données réelles, étayant chacune des contributions, sont présentés et illustrent la réalisation des deux principaux objectifs.

Mots clés : Filtre de Kalman, cartographie, SLAM, localisation, traitement de l'image, perception multi-sensorielle, robotique mobile, fusion de données, reconstruction de trajectoire.

Abstract

In the context of outdoor mobile robotics, concepts of localization and perception are central to any achievement. Also, the work in this thesis improves an existing localization process more robust without significantly increasing their complexity. The proposed problematic addresses a robot in a potentially dangerous field with the aim to follow a path established as safe with a map as simple as possible. In addition, strong constraints are imposed as in the realization (inexpensive system, undetectable) as in the result (a real-time process execution and a localization continuously within a tolerance of 10 cm closed to the reference trajectory). The exteroceptive sensor chosen to carry this project is a camera while the pose estimation of the vehicle at each moment is achieved with an Extended Kalman filter. The main estimation problems are due to the non-linearity of the models and contributions provide some solutions:

- an exact calculation method of the propagation of uncertainties in space world into space sensor (camera);
- a method to detect the main event of a divergence of the update step of the Kalman filter;
- a method to correct the Kalman gain.

This project had two objectives: to achieve a localization function with respect to strong constraints previously mentioned, and allow a vehicle to leave temporarily the reference trajectory, while the operator modify the robot trajectory and then resume the normal course of its mission to the reference path. This second part involves a broader context in which it is necessary, in addition to the localization, to map the environment. This problem, identified by the acronym SLAM (Simultaneous Localization And Mapping), made the connection with the last two contributions of this thesis work:

- an initialization method of the points which will constitute the SLAM map;
- a method to maintain consistency between the reference map and the SLAM map.

Results on real data, supporting each contribution, are presented and illustrate the realization of the two main objectives.

Keywords: Kalman filter, mapping process, SLAM, localization, image processing, multi-sensor perception, mobile robotic, data fusion, trajectory reconstruction.

Table des matières

Introduction	1
La robotique	1
La robotique mobile	2
Présentation du projet	3
Le projet RINAVEC	5
Annonce du plan	6
1 Analyse de l'existant	9
1.1 Les possibilités matérielles	10
1.1.1 Solution GPS	10
1.1.2 Solution caméra(s)	14
1.1.3 Solution par télémétrie	16
1.1.4 Rappel des contraintes et choix	17
1.2 Type de cartes	18
1.2.1 Cartes de type <i>points</i>	18
1.2.2 Cartes de type <i>amers structurés</i>	19
1.2.3 Cartes de type <i>hétérogène</i>	19
1.2.4 Bilan relatif aux types de cartes	20
1.3 Les possibilités algorithmiques	20
1.3.1 La localisation	20
1.3.2 Le SLAM	29
1.3.3 Bilan	29
1.4 Choix et conséquences	30
1.4.1 Choix de la carte	30
1.4.2 Choix de l'approche	31
1.4.3 Verrou technologique	31
2 Localisation	33
2.1 Introduction	34
2.2 Rappels sur le filtre de Kalman	34
2.3 Localisation en ligne	35
2.3.1 Prédiction de l'état du véhicule	35
2.3.2 Prédiction de l'observation	40
2.3.3 Recherche de l'observation - Association	49
2.3.4 Mise à jour	59
2.3.5 Résultats	61
2.4 Localisation initiale	62
2.4.1 Problème de l'initialisation	62
2.4.2 Solution envisagée	63
2.4.3 Résultats de localisation initiale	65

3	SLAM par filtrage de Kalman étendu	69
3.1	Introduction	70
3.2	Initialisation de nouveaux points 3D	71
3.2.1	Extraction de points	71
3.2.2	Création du point 3D	76
3.2.3	Création de la matrice d'incertitude associée	77
3.3	Prédiction	86
3.4	Mise à jour	88
3.4.1	Mise à jour avec la carte de référence	88
3.4.2	Mise à jour avec la carte SLAM	89
3.5	Problème de linéarisation du modèle d'observation	89
3.5.1	Cas d'étude 2D : $\mathbf{z} = y/x$	89
3.5.2	Calcul de la correction	93
3.5.3	Résultat de la correction	95
3.6	Retour au Visual SLAM	96
3.7	Conservation d'un point	100
3.8	Résultats de simulation	102
3.9	Conclusion	104
4	SLAM bi-monoculaire	105
4.1	Introduction et rappels	106
4.2	Relations d'observation et d'évolution	107
4.2.1	Relations d'observation	107
4.2.2	Relations d'évolution	109
4.3	Résultats	110
4.3.1	Résultats de localisation d'un point	110
4.3.2	Résultats d'autocalibration	112
4.4	Conclusion	114
5	Expérimentations	117
5.1	SLAM sur le site PAVIN	118
5.1.1	Définition des zones de recherche	119
5.1.2	A10 : SLAM sans aucune amélioration	121
5.1.3	A11 : SLAM avec l'initialisation « monde plan »	121
5.1.4	A12 : SLAM avec l'initialisation « monde plan » et la correction des zones de recherche	121
5.1.5	A13 : SLAM avec l'initialisation « monde plan », correction des zones de recherche et correction du gain de Kalman	122
5.1.6	Comparatif des algorithmes	123
5.2	Localisation à partir d'une carte de référence sur le site PAVIN	124
5.3	SLAM avec carte de référence	124
5.4	Conclusion générale des expérimentations	127

Conclusions et perspectives	131
Conclusions	131
Perspectives	133
A Le lieu des points d'un ellipsoïde n'englobant pas l'origine et dont les plans tangents contiennent l'origine	137
A.1 Lemme de commutativité matricielle	137
A.2 Le lieu des points de l'ellipsoïde dont les plans tangents contiennent l'origine est une ellipse	138
A.2.1 Plan tangent à \mathcal{E}	138
A.2.2 Plan tangent à \mathcal{E} passant par l'origine	139
B Projection d'un ellipsoïde dans une image : calcul des tangentes limites	141
B.1 Introduction	141
B.2 Equation principale du problème	141
B.2.1 Plan tangent à \mathcal{E}	141
B.2.2 Plan tangent à \mathcal{E} passant par l'origine	142
B.2.3 Recherche des plans limites	143
B.3 Pose des systèmes à résoudre	143
B.4 Résolution des systèmes	144
B.4.1 Résolution pour les points à plans verticaux	144
B.4.2 Résolution pour les points à plan horizontaux	147
B.5 Conclusion	148
C Extraction d'une ellipse 2D à partir de deux paires de points aux tangentes parallèles deux à deux	149
C.1 Introduction	149
C.2 Problématique	149
C.3 Calcul et résolution	150
C.3.1 Systèmes solutions	150
C.4 Conclusion	151
D Estimation de la profondeur d'initialisation des points 3D par une approche « monde plan »	153
D.1 Introduction	153
D.2 Problématique	153
D.3 Résolution	154
D.4 Conclusion	154
E Construction contrôlée d'ellipsoïde	155
E.1 Méthode des jacobiennes	155
E.1.1 Introduction	155
E.1.2 Problématique	155
E.1.3 Résolution	155

E.1.4	Cas particulier	156
E.1.5	Conclusion	156
E.2	Méthode de la projection d'ellipsoïde	157
E.2.1	Introduction	157
E.2.2	Problématique	157
E.2.3	Résolution	158
E.2.4	Conclusion	158
Bibliographie		161

Table des figures

1	Environnements et mobilité	2
2	La Tortue de G. Walter	2
3	Environnements de mobilité terrestre	3
4	Schéma du traitement de l'information	5
1.1	Système GPS	11
1.2	Problèmes associés au système GPS	11
1.3	Angles et orientation de repère	13
1.4	Séréoscopie	14
1.5	Les nappes laser (capteur SICK)	16
1.6	Capteur Velodyne	17
1.7	La triangulation	21
1.8	La géométrie épipolaire	22
1.9	L'ajustement de faisceaux	23
1.10	Méthode itérative versus méthode globale	27
1.11	Parabole	27
2.1	Modèle d'évolution	36
2.2	Le repère véhicule	36
2.3	Trajectoires - Simulation et reconstruction odométrique	37
2.4	La notion d'obstacle	40
2.5	Définition des variances des bruits de processus	41
2.6	Couple point-observation	43
2.7	Capteur CCD Kodak	45
2.8	Modèle pinhole d'une caméra	45
2.9	Définition des axes du plan image	46
2.10	Représentation des différents repères	46
2.11	Projection dans une image	47
2.12	Descripteur de point	50
2.13	Projection d'un ellipsoïde	51
2.14	Zone de recherche - méthode des jacobienues	51
2.15	Sous-estimation de la zone de recherche	52
2.16	Zone de recherche - cadre d'étude	56
2.17	Zones de recherche - Comparaison des méthodes	57
2.18	Schéma du processus de la localisation en ligne	60
2.19	Simulation de trajectoire	62
2.20	Précision et intégrité de la localisation	63
2.21	Redondance d'un descripteur dans une image	64
2.22	Schéma du processus de la localisation initiale	65
2.23	Montage du banc stéréo	66
2.24	Résultat d'initialisation sur images réelles	67

2.25	Localisation simultanée de deux caméra	68
3.1	Extracteur de Harris - méthodes globale et découpée	74
3.2	Extracteur de Harris - cas d'échec de la méthode globale	75
3.3	Un point 3D en coordonnées sphériques	76
3.4	Un point 3D en coordonnées cartésiennes	77
3.5	Initialisation de la covariance d'un point par le filtre de Kalman . . .	79
3.6	Schéma d'incertitude angulaire	80
3.7	Variation de l'incertitude image en fonction de l'angle d'observation .	81
3.8	Courbe de définition d'un incertitude image constante	82
3.9	Initialisation de la covariance d'un point par les jacobienness	83
3.10	Projection de l'ellipsoïde nominal dans les plans Oxz et Oxy	84
3.11	Initialisation de la covariance d'un point par les tangentes	85
3.12	Superposition des résultats d'initialisation des covariances	85
3.13	Reprojection d'une erreur image dans le monde	86
3.14	Cas d'étude - scénario	90
3.15	Cas d'étude - initialisation	90
3.16	Nouvelle position de l'observateur.	91
3.17	Mise à jour traditionnelle par le filtre de Kalman	91
3.18	Mise à jour dans un espace linéaire	92
3.19	Mise à jour à erreur d'observation variable	93
3.20	Cas d'étude - mise à jour corrigée	96
3.21	Correction de la mise à jour de Kalman - cadre d'étude 3D	97
3.22	Résultats - Avec et sans correction du gain de Kalman	98
3.23	Environnement d'acquisition PAVIN	99
3.24	Comparaison de SLAM avec/sans correction	100
3.25	Distance de Mahalanobis	101
3.26	Environnement simulé d'échappement de trajectoire	102
3.27	Résultat de la localisation sans carte SLAM	103
3.28	Résultat de la localisation avec carte SLAM	103
4.1	Définition des repères de chaque caméra.	107
4.2	Montage du banc stéréo utilisé dans une approche bi-monoculaire . .	109
4.3	Image de première observation	111
4.4	Résultat de l'approche monoculaire	111
4.5	Image de première observation	112
4.6	Résultat de l'approche Bi-monoculaire : convergence du point	112
4.7	Résultat de l'approche Bi-monoculaire : convergence de la caméra . .	113
4.8	Comparatif de l'évolution des incertitudes	113
4.9	Résultat de l'approche Bi-monoculaire : convergence des paramètres de translation de la caméra	114
4.10	Résultat de l'approche Bi-monoculaire : convergence des paramètres de rotation de la caméra	115
4.11	Courbes de convergences (odométrie fortement bruitée)	116

5.1	Trajectoire de référence dans l'environnement PAVIN	119
5.2	Zones de recherche avec et sans correction	120
5.3	Résultats de l'algorithme AL0	121
5.4	Résultats de l'algorithme AL1	122
5.5	Résultats de l'algorithme AL2	122
5.6	Résultats de l'algorithme AL3	123
5.7	Superposition des trajectoires du SLAM et de localisation	125
5.8	Ecart entre les trajectoires du SLAM et de localisation	125
5.9	Trajectoire secondaire dans l'environnement PAVIN	126
5.10	Résultat de localisation sur un parcours avec déviation	126
5.11	Résultat de localisation sur un parcours avec déviation	127
5.12	Courbes de variances du biais	128
5.13	Comparaison des courbes d'écart à la trajectoire de référence	128
C.1	Schématisation du problème.	149
D.1	Schéma du problème de calcul de la profondeur d'initialisation	154
E.1	Schéma de la problématique	157

Liste des tableaux

1	Tableaux de récapitulation des contraintes	4
1.1	Sources d'erreurs du système GPS	12
1.2	Comparaison des résultats de chaque méthode	28
1.3	Complexité de l'algorithme EKF	28
2.1	Comparaison des résultats de chaque méthode : critère de choix	44
2.2	Données de simulation	61
2.3	Tableaux de comparaison des résultats	61
3.1	Tableaux des principaux extracteurs de points d'intérêt.	72
3.2	Tableau comparatif des SLAM	99
3.3	Tableaux des incertitudes de seuil pour une référence de 10 cm ³ à 1m. .	101
5.1	Tableau récapitulatif des algorithmes de SLAM testés	119
5.2	Tableau comparatif des algorithmes SLAM	123

Introduction

La robotique

La robotique est l'ensemble des études et des techniques de conception et de mise en oeuvre des robots. Il s'agit d'un domaine multidisciplinaire qui regroupe des aspects concernant la mécanique, l'informatique ou encore l'électronique. En poussant le concept à son paroxysme, on peut considérer que la robotique est une somme de savoirs permettant de fusionner les avancées de différentes disciplines afin de fournir à l'homme des outils de plus en plus complexes tout en conservant une simplicité d'utilisation. La plus flagrante analogie de cette évolution a pour finalité le domaine médical. Dès les temps préhistoriques, l'Homme a eu besoin de se nourrir et la taille du silex lui a permis de trancher les chairs. Les évolutions faisant et l'apparition de la médecine ont engendré des couteaux de précision afin d'atteindre les lésions sans générer un surplus de dégâts : le but était donc de réduire la taille des instruments sans perdre la précision du geste et donc la maniabilité de l'outil.

Les avancées au sein des capteurs et la miniaturisation des composants électroniques ont, à leur tour, permis d'assister le chirurgien dans des opérations toujours plus complexes avec de moins en moins de dégâts. Faire remonter la robotique jusqu'à l'âge de pierre est peut-être une extrapolation litigieuse, cependant ce sont bien les avancées dans chacun de ces domaines qui, au travers de leur regroupement, ont donné naissance à la robotique. Historiquement, le terme de robotique vient du tchèque "*Robota*" qui peut être traduit par *servitude* et fait son apparition en 1920, dans la pièce de théâtre "**R. U. R.**" (Rossum's Universal Robots) de Karel Čapek. Ce terme sera rendu ensuite célèbre en 1950 par Isaac Asimov et les fameuses trois lois de la robotique qui doivent empêcher les robots de prendre le pouvoir ou de s'en prendre aux êtres humains tout en leur conférant le plus d'actions autonomes possibles.

D'une manière plus concrète, on ne conçoit pas une machine à laver ou un réfrigérateur comme un robot. Pour qu'un aspirateur entre dans la catégorie des robots, il faudra attendre 2002 qu'il puisse nettoyer une pièce sans l'aide d'un être humain. Concernant l'exemple de la machine à laver, on la considérerait plus facilement comme un robot si elle récupérait le linge de la corbeille, le triait d'elle-même, sélectionnait le programme, puis, après séchage, le pliait et le rangeait. Finalement, au-delà de la fusion de diverses disciplines, la doxa reconnaît une robotisation au pourcentage d'une chaîne d'action réalisée de manière autonome par un appareil. Les enjeux de la robotique, qui reconnaît un lave-linge comme un robot, sont de fournir une assistance à l'être humain pour des opérations complexes, dégradantes ou salissantes, et qui peuvent mettre l'Homme en danger (les anglais parlent des 3D pour *Dull*, *Durty* and *Dangerous*). Dans le cas de ma thèse, comme dans le celui qui a fait passer l'aspirateur *Roomba* dans la classe des robots pour le commun des mortels, c'est à la notion de mobilité que nous allons nous intéresser.

La robotique mobile

Dans le règne animal et tout particulièrement chez les mammifères régis par le cycle alimentaire, un être qui ne marche pas dans les premières minutes de son existence voit son espérance de vie fortement diminuée. Les humains possèdent, dès leur naissance le réflexe de la marche automatique. La condition de mobilité est une qualité quasiment inhérente à la vie, conférant autant d'autonomie que de sources de dangers tout aussi prévisibles qu'impromptues. Les environnements sont aussi multiples que les façons de se mouvoir (voir figure 1a et 1b) et nous allons porter notre attention sur les robots mobiles terrestres. La communauté s'accorde à

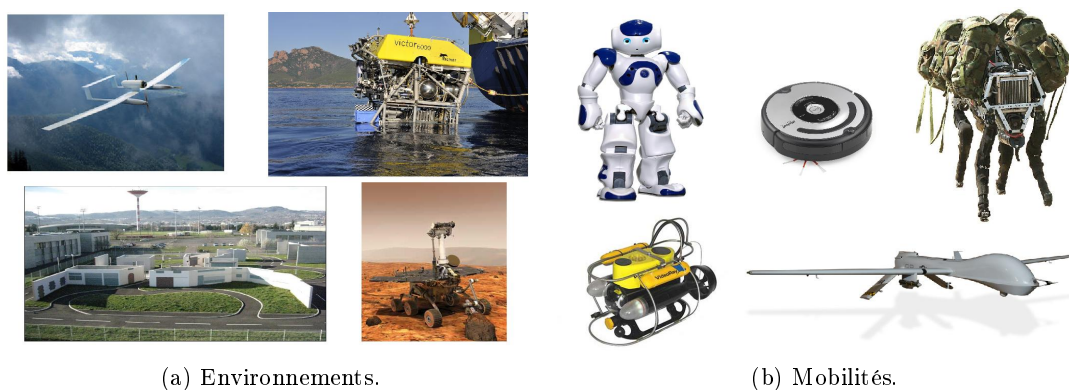


FIGURE 1 : **1a** : Illustration non exhaustive d'environnements explorés par la robotique. **1b** : Illustration non exhaustive de différentes techniques de mobilité (bipède, quadrupède, à roue, à élice ou encore à réaction).

reconnaitre la **Tortue** réalisée en 1950 par Grey Walter et surnommée Elsie comme sans doute le premier robot mobile (Figure 2). L'autonomie de déplacement est

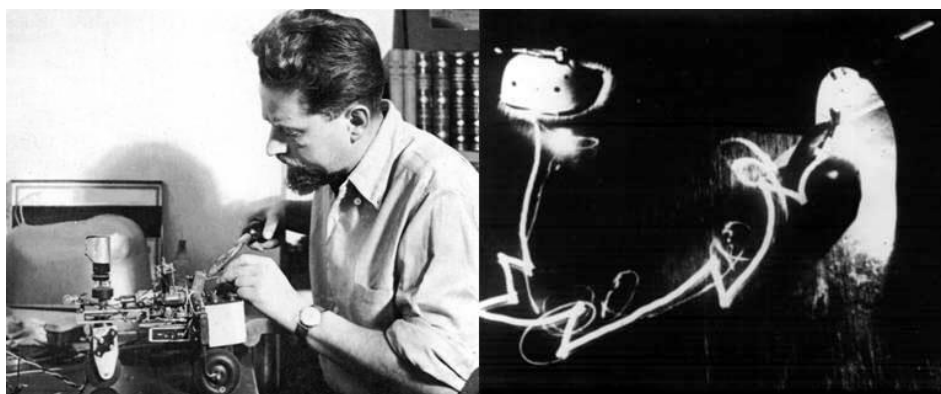


FIGURE 2 : G. Walter et sa *Tortue Elsie* et une image de la trajectoire du robot jusqu'à sa niche

un problème complexe dans l'interaction avec les humains. Faire se déplacer un

véhicule à 120 km/h de manière autonome est chose aisée. S'assurer dès lors, de la conservation de son intégrité ou de celle de son environnement donne toute la difficulté au problème de mobilité.

Sans aller cependant jusqu'à de telles vitesses, la mobilité des outils, et donc leur robotisation, offre plus de sécurité, de disponibilité ou d'accessibilité à certains sites. Ainsi, envoyer un robot opérer une soudure au coeur d'un réacteur nucléaire, permettre à un chirurgien d'effectuer une opération à l'autre bout du monde avec une précision bien supérieure à celle de la main d'un homme ou encore de ramener des échantillons de Mars ou de la Lune n'ont été réalisables que grâce à la possibilité de se mouvoir.

Finalement, l'ouverture au monde au travers de la mobilité impose bien souvent de correctement définir la zone d'action du robot. Les problèmes perçus sont croissants avec la diversité de l'environnement rencontré et le nombre d'interactions pouvant intervenir au cours de l'exécution d'une mission (la figure 3 montre différents environnements considérés en robotique mobile terrestre).



FIGURE 3 : Environnements de mobilité terrestre : milieu urbain, milieu à mobilité réduite (pyramide), milieu désertique et milieu rural.

Cadre des travaux de Thèse

Le cadre des travaux de thèse nous place dans un environnement potentiellement dangereux. Au sein d'une zone inconnue, un véhicule doit se déplacer dans un couloir de navigation préalablement établi (par exemple par un premier véhicule), afin

de garantir son intégrité ainsi que celle de son équipage ou de son chargement. Le couloir de navigation définissant une zone sans danger, il convient cependant d'avoir le minimum d'action envers l'environnement afin de garantir au maximum une non-déteçtabilité (cela inclut toute émission d'un quelconque signal, qu'il permette à un capteur d'acquérir une information sur l'environnement ou bien de transmettre sa localisation par exemple). De plus, ouvrir une voie dite sécurisée en terrain potentiellement dangereux de grande largeur entre en conflit avec la notion de non-déteçtabilité. Enfin, le projet aspire à une réalisation la moins chère possible, afin de ne pas être un frein au développement des futures solutions envisagées. Il ne s'agit pas là d'une contrainte forte : une des conclusions de l'étude pourrait être que la fonction désirée ne peut être obtenue qu'avec des capteurs onéreux.

Cette problématique n'est pas le but premier du projet qui considère comme acquise une carte 3D de l'environnement ainsi que le couloir de navigation : l'objectif premier est de pouvoir rejouer un chemin de manière temps réel tout en restant dans le couloir de navigation. La carte est une image de l'environnement dans laquelle le véhicule se localise et doit être définie. Les amers qui constituent la carte peuvent être de différentes natures tandis que le couloir de navigation est défini comme une bande centrée par rapport à la trajectoire principale. En terme d'écart à la trajectoire réalisée par le processus en charge d'ouvrir la voie sécurisée, le projet propose une contrainte de 10 cm sur une distance pouvant dépasser plusieurs kilomètres. Il s'agit bien ici d'une contrainte de la trajectoire du système à réaliser par rapport à celle du système en charge d'ouvrir la voie et non d'une contrainte vis-à-vis de la position absolue de l'un ou de l'autre des processus.

La dernière contrainte se veut être une recommandation, à savoir une quantité d'information relativement faible, permettant de stocker, de télécharger ou de mettre en place rapidement les données nécessaires à la réalisation du processus de jeu.

Ces contraintes, rappelées dans le tableau 1, sont en majeure partie celles du projet RINAVEC¹ dont l'objectif est de réaliser un convoi en zone à risque avec une distance inter-véhicules suffisamment variable pour autoriser la perte de vue du véhicule précédent. Dès lors qu'on se place dans le cadre de la robotique où l'on

Entités	Position	Déteçtabilité	Coût	Environnement	Distance
Contraintes	10 cm	Minimale	Faible	Ouvert/Accidenté	1 km

TABLE 1 : Tableaux de récapitulation des contraintes

cherche à fournir une autonomie à un robot vis-à-vis d'une certaine catégorie d'évènements susceptibles d'arriver, on se confronte au schéma récurrent **Perception - Décision - Action** (voir figure 4). Pour une perception donnée, on choisit une décision. Cette décision, avant même d'engendrer une action peut être validée par une information complémentaire ne faisant pas partie de la perception initiale. La reconnaissance de cette information complémentaire (ou non) permet de valider (ou d'invalidier) le processus décisionnel qui donnera lieu à une action. La validité de

1. RINAVEC : Reconnaissance d'Itinéraire et Navigation en convoi de Véhicules Communicants

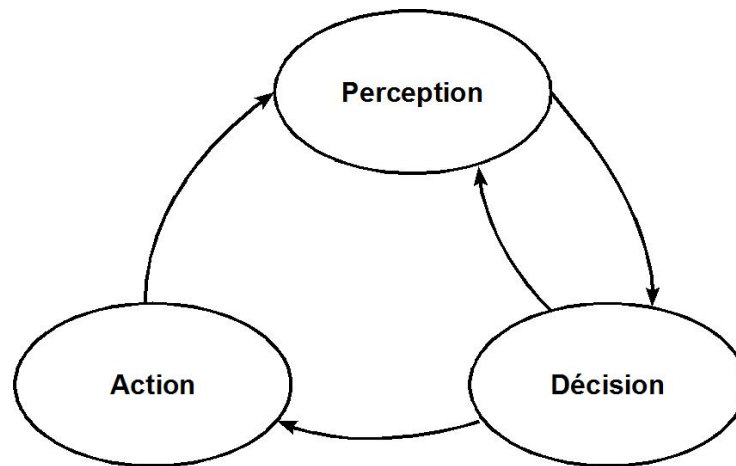


FIGURE 4 : Démarche du traitement de l'information

l'exécution de l'action ne se fera que par la perception des conséquences mesurables qu'elle implique.

Sans avoir défini ni abordé le type de perception utilisé pour réaliser la fonction de rejeu ou les aspects de commande à mettre en place pour le déplacement effectif du véhicule, il est évident que le problème est avant tout un problème de **localisation**. C'est notamment en connaissant notre position par rapport au chemin de référence que l'on pourra correctement estimer où aller. Le système étant destiné à différents types de véhicules, sans plus de précision sur leur capacité de mouvement, c'est l'expression d'un processus de localisation robuste et tenant compte au maximum des contraintes qui sera développée.

Le projet RINAVEC

Ce projet, dont la thématique scientifique centrale est le SLAM (Simultaneous Localization And Mapping) combinant perception et estimation de l'état des véhicules et de l'environnement, est proposé par trois partenaires : d'une part des équipes du LAAS-CNRS et du LASMEA, expertes dans les domaines de Robotique Terrestre ou Véhicules Intelligents, et d'autre part, le département « Robotique et UAV » de Thales Optronique S.A., réputé pour sa capacité à intégrer des systèmes robotiques. Il est soutenu par l'ANR-PsiRob (Agence Nationale pour la Recherche - Programme Systèmes Interactifs et Robotique) et a débuté en avril 2008 (Rinavec PsiRob - ANR-07-ROBO-0006).

RINAVEC vise à développer et évaluer des fonctions avancées de Perception et de Modélisation de l'Environnement, pour des véhicules évoluant en convoi sur un itinéraire inconnu a priori, en milieu ouvert (péri-urbain et naturel). Le contexte applicatif concerne la navigation de convois humanitaires dans des zones de conflit ou accidentées : pour diverse raisons (mines, autres véhicules ou personne), la progression de tels convois est très difficile et dangereuse. Le but sera d'évaluer les progrès

en Perception pour la navigation de véhicules communicants dans ce contexte très exigeant.

Chaque véhicule du convoi est équipé de capteurs proprioceptifs bas-coût (GPS, inertiel, odométrie...) et de caméra. Le véhicule de tête (leader) modélise la trajectoire suivie par le convoi ; pour éviter tout danger, les autres véhicules (suiveurs) doivent emprunter cette trajectoire, avec une déviation maximale de 10 cm. Le leader, télé-opéré par un opérateur qui peut être dans un véhicule du convoi, enregistre sa trajectoire sous la forme de positions successives, relatives à des amers 3D fixes de l'environnement, détectés et suivis dans les images acquises en mouvement. Trajectoires et positions des amers fixes sont stockés dans une carte, périodiquement envoyée aux suiveurs. Chacun l'exploite pour se localiser par rapport aux amers déjà perçus par les précédents, et pour corriger sa trajectoire afin de respecter la contrainte de déviation maximale.

Plusieurs variantes de ce scénario seront considérées : convoi espacé (les véhicules ne se voient pas) ou resserré (localisation mutuelle), rejeu différé d'un itinéraire, véhicule sans capteur guidé par les autres.

Annnonce du plan

Le projet se traduit dans un premier temps par la réalisation d'un processus de rejeu de trajectoire sous contraintes, dans un couloir de navigation défini au sein d'une carte. La carte, la trajectoire ainsi que les contraintes déterminant notamment le couloir de navigation étant pré-établies, il va falloir trouver un ensemble de capteurs et une méthode permettant une localisation suffisante pour tenir la contrainte de positionnement.

Une étude préalable a été menée sur les différents capteurs pouvant répondre à la problématique. De plus, chaque capteur, associé à une méthode, offre son lot de solutions comme son lot de contraintes inhérentes. L'étude et les choix qui ont été faits pour la réalisation de la fonction de localisation fait l'objet du chapitre 1.

La conclusion de cette analyse aboutissant à la sélection des capteurs et des méthodes, la réalisation du système fera l'objet du chapitre 2. Le comportement global de l'algorithme ainsi que le cas particulier de l'initialisation y sont décrits. De plus, un premier verrou théorique concernant l'observation sera levé.

Dans l'optique de rendre le système encore plus autonome, considérer que l'environnement n'a pas évolué entre les passages de chaque véhicule est en soit une contrainte. Elle ouvre sur une problématique complémentaire : comment se détourner de la trajectoire de référence sans pour autant perdre la localisation ? Par ailleurs, cette modification de la trajectoire de référence obligera à revenir sur la trajectoire principale et d'effectuer un raccordement pour reprendre la fonction de rejeu traditionnelle. Cette problématique, développée dans le chapitre 3, offre aussi une solution pour la construction de la carte que l'on qualifiera de référence (construite par un premier véhicule ouvrant la voie sécurisée par exemple).

Le chapitre 4 fait référence à une solution d'optimisation découlant directement

de l'analyse des capteurs du chapitre 1. Nous verrons alors comment s'appliquent les différentes corrections et résolutions développées dans les chapitres précédents.

Toutes les études étant réalisées sur des modèles de simulation, le chapitre 5 présente l'application sur des données réelles en les comparant aux différentes contraintes du cadre de travail.

Analyse de l'existant

Sommaire

1.1	Les possibilités matérielles	10
1.1.1	Solution GPS	10
1.1.2	Solution caméra(s)	14
1.1.3	Solution par télémétrie	16
1.1.4	Rappel des contraintes et choix	17
1.2	Type de cartes	18
1.2.1	Cartes de type <i>points</i>	18
1.2.2	Cartes de type <i>amers structurés</i>	19
1.2.3	Cartes de type <i>hétérogène</i>	19
1.2.4	Bilan relatif aux types de cartes	20
1.3	Les possibilités algorithmiques	20
1.3.1	La localisation	20
1.3.2	Le SLAM	29
1.3.3	Bilan	29
1.4	Choix et conséquences	30
1.4.1	Choix de la carte	30
1.4.2	Choix de l'approche	31
1.4.3	Verrou technologique	31

1.1 Les possibilités matérielles

L'aspect principal de la réalisation d'une fonction de jeu réside dans la tâche de localisation à mettre en place. Dans le cadre du projet, deux options de localisation sont à aborder : une localisation en vue d'une simple navigation dans une carte où tout est pré-établi, et une localisation avec enrichissement personnel où le véhicule pourra rajouter des informations. A titre d'exemple, lorsque l'opérateur souhaite le faire dévier de sa trajectoire initiale pour éviter un obstacle par exemple.

La littérature offre différentes méthodes de réalisation d'un processus de localisation en fonction des capteurs et des contraintes. Dans un premier temps, l'analyse des possibilités, des avantages et inconvénients de chaque capteur, associés aux solutions proposées dans la littérature, nous permettra de faire un choix en rapport avec les contraintes du cadre d'étude.

Dans un deuxième temps, nous étudierons les informations nécessaires à un véhicule pour se localiser (celles-ci proviennent, par exemple, d'un premier véhicule ayant défini, entre autre, la trajectoire et donc le couloir de navigation). Ces informations, outre celles relatives à la trajectoire à suivre, formeront ce que l'on appellera par la suite **la carte**. Enfin, et sur la base du ou des capteurs choisis, nous étudierons les quelques algorithmes principaux, récurrents dans ce type de travaux de localisation. De la même manière que pour les capteurs, nous devons faire un choix quant à la méthode à employer.

1.1.1 Solution GPS

Dans les approches de localisation, le premier capteur auquel on fait référence est bien entendu le capteur absolu de prédilection dans le domaine : le **GPS** (Global Positioning System). Créé dans les années 1970 par le département de la défense des États-Unis, il est à ce jour sans doute le plus populaire. Une constellation de plusieurs satellites synchronisés transmettent en continu leur position datée. Un calcul de triangulation à partir des signaux reçus permet de localiser un récepteur sur le globe (voir figure 1.1).

L'utilisation civile du GPS fournit une position avec une précision de l'ordre d'une dizaine de mètres [Hurn 1998]. Les systèmes GLONASS ou GALILEO que l'on retrouve sous l'acronyme GNSS (Global Navigation Satellite System) entrent aussi dans la catégorie des GPS. L'imprécision générale du GPS peut être réduite par l'utilisation de corrections diffusées globalement pour toute une région du globe comme le système EGNOS (European Geostationary Navigation Overlay Service) pour l'Europe, le système WAAS (Wide Area Augmentation System) pour le nord de l'Amérique ou encore le système MSAS (Multi-functional Satellite Augmentation System) pour l'Asie.

Toujours dans le domaine civil, l'ambiguïté de position est considérablement réduite grâce à la mise en correspondance avec le tracé des routes. Dans le cadre professionnel, on a souvent recours à la version **GPS-RTK** (Real Time Kinematic). Une base, dont la position est parfaitement connue, fournit des corrections de

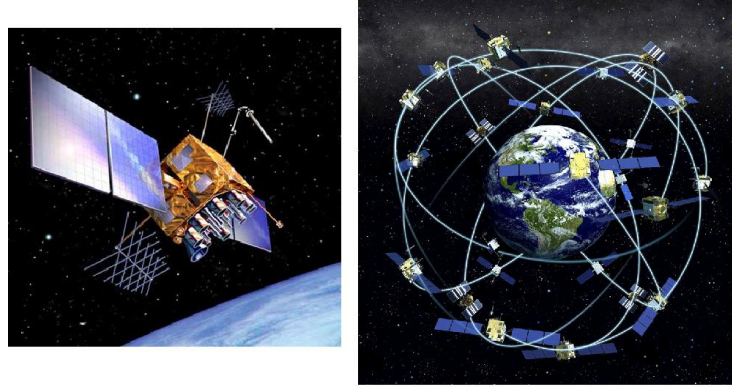
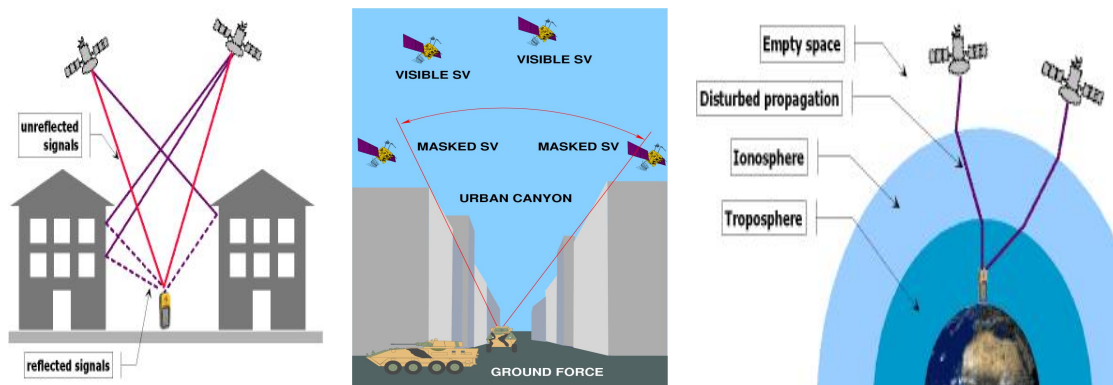


FIGURE 1.1 : Un satellite du système GPS-NAVSTAR et une représentation de la constellation de satellites autour de la Terre.

position permettant à un récepteur, dans le champs d'action de la base, d'obtenir une précision centimétrique. Le système GPS résout le problème de la localisation d'un récepteur de manière absolue. Il présente cependant quelques inconvénients que nous allons aborder ici. Le calcul de position se fonde sur le temps de vol du signal supposé reçu directement. Si, pour une raison ou une autre, le signal est perturbé, alors la position calculée est dégradée. Le cas de multi-réflexion du signal en est l'expression la plus évidente. Un signal, considéré comme reçu directement, arrive jusqu'au récepteur après s'être réfléchi sur une paroi. Le surplus de trajet se répercute dans le calcul de la position et dégrade le résultat (voir figure 1.2a). De plus, les hauts immeubles constituent des obstacles à l'observation des satellites émetteurs des signaux GPS : on parle alors de canyons urbains (voir figure 1.2b). Enfin, les différentes couches de l'atmosphère engendrent à leur tour des modifications du trajet du signal (voir figure 1.2c).



(a) Multi-réflexion du signal (b) Problème du canyon urbain. (c) Impact des couches atmosphériques. GPS.

FIGURE 1.2 : Problèmes associés au système GPS.

De plus, bien que les satellites aient été disposés dans l'espace de manière à ce

qu'une constellation suffisante soit observable, le nombre de satellites disponibles est parfois insuffisant. Le système doit résoudre un problème comptant 4 variables (3 pour la position et 1 pour le temps). Cependant, les différentes erreurs appliquées au signal ne sont déduites que par la redondance d'information. C'est pourquoi, 7 satellites observables (au lieu de 4 théoriquement) sont considérés comme une constellation suffisante. Les provenances des principales sources d'erreurs du système GPS sont rappelées dans le tableau 1.1. Bien que celles-ci soient complexes,

Source	Effet
Satellites	Dérive d'horloge Erreur orbitale
Propagation du signal	Réfraction ionosphérique Réfraction troposphérique
Récepteur	Dérive d'horloge Multi-trajet du signal Variation de l'origine de phase de l'antenne canyon urbain zone de masquage

TABLE 1.1 : Source d'erreurs et de biais du signal.

inhérentes et que leur correction reste encore un problème d'actualité, un système de localisation peut tout à fait être développé sur ce capteur. [Lenain 2005] propose une réalisation des fonctions de localisation et de guidage dans le domaine agricole en champ dégagé.

L'utilisation d'un modèle de déplacement du véhicule, fondé notamment sur des capteurs proprioceptifs, permet de connaître une estimation de la position dans le temps d'indisponibilité du signal GPS. Les capteurs proprioceptifs conduisent à des localisations locales et non plus absolues. Ces données, associées à un modèle cinématique (ou dynamique) d'un véhicule permettent d'estimer le déplacement de celui-ci. [Campion 1996] associe au modèle cinématique d'un véhicule les informations odométriques (distance curviligne parcourue par une roue) ainsi que les informations d'un gyromètre (capteur de vitesse angulaire) pour estimer la position d'un véhicule.

D'une manière générale, on doit faire face à un manque de précision mais aussi à un défaut de disponibilité du signal voire des données non entières (cas des multi-trajets par exemple).

Par ailleurs, même s'il est vrai qu'en étudiant les positions successives uniquement obtenues par le GPS, il serait théoriquement possible de remonter à l'information de cap (ou lacet) du mobile, ce capteur ne permet pas de déterminer les deux autres angles à savoir le tangage et le roulis (voir figure 1.3). De la même manière, ces deux derniers angles ne peuvent être obtenus avec l'odométrie seule couplée au GPS. La fusion de données avec une centrale inertielle, capteur à l'image de l'oreille interne de humaine, comble ce manque d'informations. Une centrale fournit des informations d'accélération et de vitesse suivant les 3 axes permettant de remonter aux

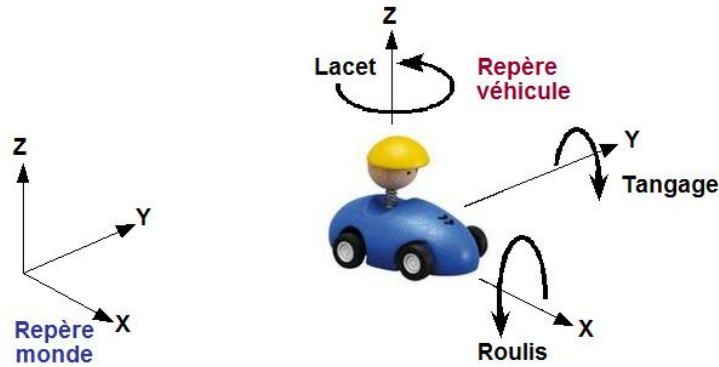


FIGURE 1.3 : Le roulis, le tangage et le lacet permettent de définir de manière complète et unique l'orientation d'un repère dans un autre.

angles définissant l'orientation du véhicule. Ce capteur est d'une précision proportionnelle à son prix : les moins chères sont suffisamment précises pour des grandes variations d'accélération mais ne permettent généralement pas d'obtenir un système de localisation fiable et précis. Bien que [Vaganay 1993] ait fusionné 5 centrales bas-coût pour obtenir une estimation correcte de l'attitude, restons avec la fusion GPS/centrale inertielle en mentionnant les travaux de [Ryu 2004] et de [Bevly 2004] qui donnent une solution à notre problème de localisation en environnement ouvert.

Sans lui consacrer une section à part entière, le système **SIG** (Système d'Information Géographique) est, pour certains, vu comme un capteur à lui seul. Il constitue une banque d'informations topologiques de l'environnement et d'amers géo-référencés [Tessier 2007]. Couplé avec un capteur extéroceptif, une simple triangulation permet de remonter à la position recherchée. [Jensfelt 2001] utilise une caméra pour mettre en relation l'observation d'amers de l'environnement avec les informations de la base de données SIG pour positionner un robot. Finalement, le SIG se couple toujours à un autre capteur afin de mettre en relation une perception de l'environnement avec la base de données. Il est quand même à noter que le système SIG fournit un positionnement absolu dans le monde et permet de palier la perte de signal GPS, comme c'est le cas en zone urbaine dense. [Renault 2005] propose justement une utilisation conjointe du SIG avec le GPS et une caméra comme capteur complémentaire au SIG pour améliorer le processus de localisation.

Enfin, le GPS est aussi associé à des capteurs tels que les caméras dans [Agrawal 2006] ou d'autres comme le radar ou les télémètres (que l'on nommera capteurs actifs pour leur action sur l'environnement). Ils seront étudiés plus particulièrement dans les sections qui vont suivre.

En conclusion, le GPS reste le capteur de prédilection en matière de positionnement absolu. L'indisponibilité du signal GPS ou encore la non-intégrité de la donnée ont été corrigées par fusion avec d'autres capteurs, améliorant grandement les résultats de localisation. Cependant, les différentes sources d'erreurs et de dérive du GPS ne sont pas parfaitement modélisables (comme c'est le cas de la densité tropo-

sphérique) et sont, dans bien des cas, assimilés à du bruit ou directement éliminés. Finalement, à l'image de son utilisation civile, le couplage positionnement absolu et positionnement local reste le meilleur parti.

1.1.2 Solution caméra(s)

Les caméras sont généralement considérées comme étant les yeux des robots. Il en existe de toute sorte et elles nécessitent bien souvent une calibration avec un résultat de localisation fortement dépendant de cette étape. Sont tout de même à mentionner les travaux de [Xie 1997] qui réalise une fonction de suivi qualitative (sans calibration des caméras) par perception directe d'une mire placée sur l'objet à traquer. Dans la majorité des cas, la précision de la calibration intrinsèque (paramètres propres à la caméra) et extrinsèque (positionnement de la caméra dans le repère du robot) est souvent la pierre angulaire d'une localisation cohérente.

En se basant sur ce qui nous est le plus commun en terme d'observation, à savoir nos yeux, nous allons commencer par le cas stéréoscopique. La vision stéréoscopique impose d'avoir deux caméras dont les champs de vision se recouvrent (voir figure 1.4). L'avantage d'avoir deux observations simultanées d'une même zone sous deux angles

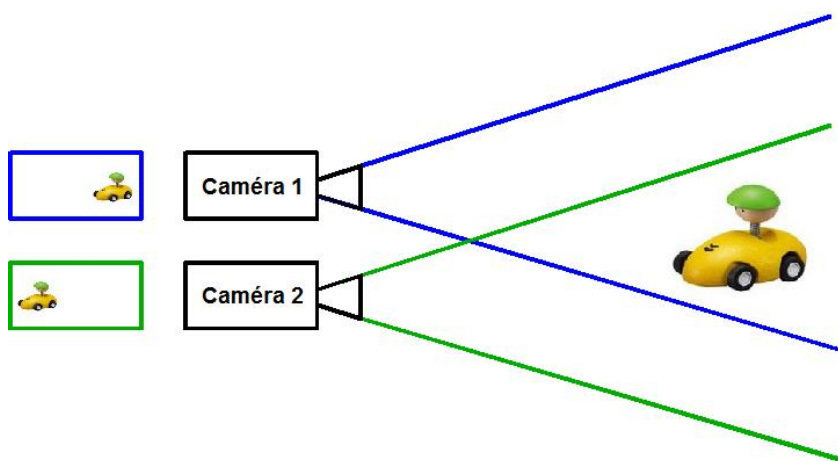


FIGURE 1.4 : Deux caméras en montage stéréoscopique observent un objet. L'observation associée à la caméra 1 est donnée dans le cadre bleu tandis que celle de la caméra 2 est donnée dans le cadre vert.

différents permet d'estimer le déplacement effectué par la paire de caméras. Ce principe définit l'odométrie visuelle où il est question d'estimer des distances par vision. L'inconvénient majeur est que la précision du résultat est très fortement liée à la précision de la calibration du banc stéréoscopique. De plus, la triangulation est conditionnée par la distance qui sépare les caméras : plus elles sont éloignées, plus la reconstruction est précise. Ceci explique notamment pourquoi, avec seulement 6 cm en moyenne entre nos deux yeux, nous avons du mal à distinguer la profondeur au-delà d'une dizaine de mètres. Pour les caméras, qui peuvent observer à l'infini à la manière des télescopes, il y a un dilemme : plus on écarte les caméras, plus le

champ de recouvrement est faible, mais plus les résultats de localisation sont précis. A contrario, plus on rapproche les caméras, plus le champ de recouvrement devient important mais des effets de parallaxe nuisent à la précision de la localisation.

La plupart des traitements en vision stéréoscopique ne se contentent pas uniquement de localiser un mobile mais estime aussi la position des amers de l'environnement. Ces deux volets sont rassemblés sous l'acronyme SLAM (Simultaneous Localization And Mapping) que nous développerons dans la section 1.3.2. Avec une approche SLAM et en vision stéréoscopique, citons [Garcia 2004] qui réalise une fonction localisation en environnement extérieur. Les résultats sont en général obtenus sur des terrains faiblement accidentés ce qui implique de faibles variations des paramètres de calibration extrinsèque. Néanmoins, [Broggi 2001] propose une solution d'auto-calibration de la paire de caméras, au cours de la mission, en utilisant des marques disposées sur le capot du véhicule.

Cette sensibilité des paires stéréoscopiques aux vibrations est connue et la synchronisation nécessaire entre les deux caméras réduit souvent les cadences nominales des caméras. Avant de passer aux approches à une seule caméra, dites monoculaires, la littérature présente des travaux sur le bi-monoculaire. Il s'agit d'une paire de caméras utilisées de manière indépendante. La deuxième caméra est considérée comme embarquée sur un autre véhicule dont le comportement est rigide par rapport au premier. L'introduction des paramètres de position de la deuxième caméra parmi les paramètres à estimer permet une souplesse au système et une robustesse supplémentaire. [Solà 2008] développe cette théorie et fournit des résultats semblables à ceux obtenus par stéréoscopie. Encore une fois ici, la fréquence des caméras est souvent supérieure à 25 Hz pour des vitesses de déplacement dépassant rarement les 2 m/s.

L'observation au travers d'une seule caméra à un instant donné ne permet pas de mesurer la profondeur d'une scène. On peut tout de même remarquer qu'en mettant en relation les observations provenant de dates différentes, le système monoculaire peut alors être considéré comme stéréoscopique dans le temps : cette approche permet théoriquement de connaître la profondeur mais l'incertitude sur le déplacement engendre une dérive. Dans le cadre monoculaire, la localisation à l'aide d'une seule caméra embarquée sur un véhicule est la principale problématique du *Bearing only*. Cette problématique se pose non plus seulement en terme de distance comme les précédentes, mais en terme d'orientation. Ainsi, bien que le cap puisse être parfaitement estimé, la position, d'un point de vue local, est souvent dégradée. Encore une fois, au sein d'un processus SLAM, les travaux de [Solà 2005] apportent une solution au problème de localisation. L'absence de mesure de la profondeur rend aussi les cartes cohérentes entre elles à un facteur d'échelle près. C'est pourquoi l'approche monoculaire, comme les précédentes d'ailleurs, est souvent couplée à un capteur de profondeur. [Malartre 2011] propose un couplage caméra et télémètre laser, fournissant en temps réel la localisation ainsi qu'un modèle numérique de terrain, évitant du coup ce problème de facteur d'échelle.

Dans tous les cas, un modèle d'évolution est défini, qu'il soit à vitesse ou accélération constante ou réalisé à partir d'un odomètre et d'un capteur angulaire. L'avantage du monoculaire réside dans son coût, sa simplicité de mise en oeuvre, sa

robustesse aux accidents de terrain et la richesse d'information fournies. Cela est à mettre en regard avec des processus plus complexes pouvant parfois nuire à l'aspect temps réel.

1.1.3 Solution par télémétrie

Terminons cette partie d'analyse des possibilités capteurs avec le cas des capteurs de distances. Dans cette catégorie, on retrouve les télémètres laser avec les nappes SICK (voir figure 1.5). [Newman 2006] montre la réalisation, encore une fois au

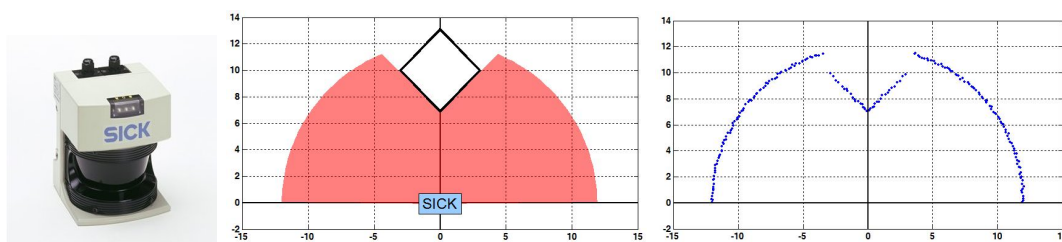


FIGURE 1.5 : **A gauche**, le capteur SICK, **au centre**, un environnement simulé et **à droite**, la réponse simulée du capteur face à cet environnement pour une distance maximale d'observation fixée à 12 unités.

travers d'un processus SLAM, d'une fonction de localisation utilisant ce capteur seul. Généralement, les nappes laser sont utilisées pour la surveillance de zones, détectant toute intrusion non autorisée. Le champ d'observation est souvent réduit à 180° , voire moins, et la superposition des nappes laser ainsi que l'ouverture d'angle de 360° ont vu le jour avec le capteur **Velodyne** (voir figure 1.6). Ce capteur étant très onéreux, rares sont les publications le concernant. Remarquons simplement qu'il est actuellement implanté sur la *GoogleCar* dont les premiers essais en mars 2011 en automatique l'ont rendue célèbre.

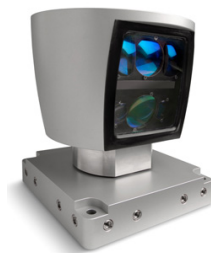
Le radar est bien souvent l'oublié du domaine. Il offre pourtant une incroyable robustesse aux différentes intempéries et autres conditions de luminosité. Son principe est simple, une onde de haute énergie est émise et le temps de vol entre son émission et la réception de son écho par une surface permet d'estimer l'environnement ou notre déplacement par rapport à celui-ci. Il permet même, dans certains cas, de voir à 360° et par delà certains matériaux ce qui est non négligeable dans le domaine de la commande et dans la phase décisionnelle. [Vivet 2010] illustre l'utilisation de ce capteur dans une application de localisation et de cartographie pour une vitesse de l'ordre de 10 m/s. Par ailleurs, pour un capteur panoramique à rotation lente (quelques hertz voir moins), des phénomènes de distorsion non négligeables apparaissent lorsque le robot se déplace. Les autres inconvénients principaux à mentionner sont sans doute ceux propres au traitement du signal (comme le multi-trajet déjà mentionné dans le cas du GPS). Le **speckle** est aussi une conséquence du multi-trajet et du multi-écho : chaque réflexion sur une surface plus réfringente que le milieu de vol entraîne un déphasage du signal qui peut, par somme avec d'autres, créer des obstacles inexistantes ou en faire disparaître.



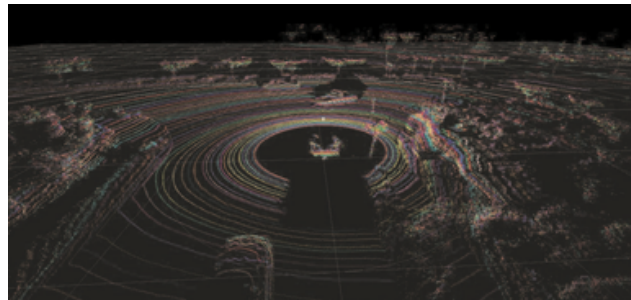
(a) Précurseur de la nappe laser 3D.



(b) GoogleCar.



(c) Capteur Velodyne.



(d) Environnement cartographié par un velodyne.

FIGURE 1.6 : **1.6a** : association de de nappes laser pour obtenir une représentation 3D de l'environnement. **1.6b** : le capteur laser 3D Vélodyne tel qu'il est monté sur la *GoogleCar*. **1.6c** : le capteur Velodyne seul, équivalent de l'intégralité du système multi-nappes de la figure 1.6a. **1.6d** : la réponse de l'observation d'un environnement extérieur par le capteur Velodyne.

Pour conclure, les capteurs de profondeur permettent une cartographie très précise et rapide de l'environnement. Cependant, outre le fait que ce sont des capteurs actifs qui laissent une empreinte visible (ce qui peut être gênant dans le cas militaire par exemple), ils restent des capteurs très onéreux pour une précision suffisante. Par ailleurs, dans le cas des capteurs à rotation lente comme le radar panoramique, un phénomène de distorsion doit être pris en compte pour corriger la position d'émission d'un rayon en amont de chaque traitement.

1.1.4 Rappel des contraintes et choix

La fusion de capteurs permet, en tenant compte des incertitudes associées à chaque capteur, de trouver la meilleure estimation de position ainsi que l'erreur correspondante. L'utilisation d'informations extéroceptives uniquement n'est pas réellement envisageable, c'est pourquoi un modèle d'évolution du véhicule sera décrit et les résultats seront fusionnés. Ce modèle d'évolution s'appuie sur des informations proprioceptives qui seront fournies par un odomètre et une mesure de l'angle de la roue directrice (ou de son équivalent). La centrale inertielle permettrait d'avoir une

mesure des angles des roulis, tangage et lacet mais son prix ne permettait plus de considérer alors le système comme bas-coût.

Suite à ce survol des différents capteurs, considérons le contexte et les contraintes du projet. Dans une première approche, une fonction de localisation doit être réalisée afin de pouvoir rejouer un chemin en zone potentiellement dangereuse. Toujours pour des raisons de sécurité, le système se veut le moins visible possible avec un maximum d'autonomie. Par conséquent, il faut travailler avec des capteurs dont l'information est continuellement disponible. De plus, le critère d'invisibilité pose aussi le principe du minimum d'interactions environnementales.

Le radar comme le télémètre, qu'il soit laser ou à ultra-son, ne semble pas convenir à la contrainte d'invisibilité, des pièges pouvant être sensibles aux ondes émises par ces capteurs.

Le signal GPS est souvent bruité dès lors que l'on entre dans une zone à risque, rendant impossible l'utilisation des trames spécifiques. De plus, disposer une base sur ce terrain pour avoir une correction centimétrique n'est pas vraiment envisageable. Cependant, bien qu'il ne puisse être utilisé comme capteur principal, il pourrait fournir une vérité terrain dans une optique de contrôle par rapport à un positionnement absolu.

Compte tenu du contexte d'évolution et des contraintes liées, notre choix s'est porté sur les caméras en terme de capteur extéroceptif. La configuration retenue a été guidée par la nature du terrain. Celle-ci pouvant être fortement accidentée, nous avons fait le choix d'une configuration monoculaire qui s'avère moins sensible à des décalibrages. La configuration bi-monoculaire offrant une robustesse supplémentaire, elle est conservée comme solution possible en cas d'une localisation insuffisante avec une seule caméra. Les capteurs d'odométrie et d'angle de roue directrice fourniront les informations proprioceptives nécessaires à la résolution du facteur d'échelle.

1.2 Type de cartes

Lors de l'introduction, nous avons mentionné, sous le nom de carte, l'ensemble des informations relatives à l'environnement qui doivent être transmises afin de pouvoir naviguer dans la voie sécurisée. Nous allons à présent venir préciser cette notion en étudiant les différents types d'informations utilisables dans des cartes.

1.2.1 Cartes de type *points*

L'information la plus couramment extraite de l'environnement est sans doute celle d'un point. Sans dimension géométrique, il est souvent associé à un descripteur qui permet sa reconnaissance dans les observations suivantes. Au-delà du descripteur et de toutes autres données qui lui seraient relatives, nous n'avons trouvé aucune publication mettant en avant une représentation du point comme meilleure qu'une autre (représentation cartésienne, polaire, sphérique, etc). Un point, dans l'espace 3D, c'est 3 coordonnées indépendantes. Se dans [Se 2001] utilise une représentation cartésienne (x, y, z) tandis que des capteurs comme les télémètres privilégient

une représentation polaire (ou sphérique) $(\rho, \theta_1, \theta_2)$. Depuis quelques temps, une représentation IDP (Inverse Depth Parametrization) en profondeur inverse semble émerger [Lemaire 2005]. Cette dernière représentation est une sur-paramétrisation du point dont l'allure de la représentation 3D de l'incertitude correspond mieux à l'aspect gaussien des bruits. Si la paramétrisation cartésienne est la plus rapide, elle pose une difficulté de représentation et de qualification des points très éloignés. Ce problème est déplacé à la localisation des points extrêmement proches par la paramétrisation en inverse de la profondeur. Il est à noter dans ce cadre que [Solà 2005] propose justement une solution à ce problème de représentation de l'incertitude en multipliant les gaussiennes.

Même si les cartes de points sont sans doute les plus répandues, il reste encore des verrous technologiques liés à leur observation, leur reconnaissance ou encore leur paramétrisation. Par ailleurs, un point et son voisinage ne sont pas parfaitement discriminants et c'est la raison pour laquelle des approches avec des amers plus structurés ont été développées.

1.2.2 Cartes de type *amers structurés*

En remarquant que notre environnement est constitué de lignes comme celle des arêtes d'un bâtiment, on est en droit de proposer des représentations plus évoluées que celle d'un simple point. [Garulli 2005] ou encore [Solà 2009] proposent, au travers d'un SLAM, l'utilisation d'une carte d'amers de type ligne. [Ohno 2005] propose de pousser la structuration jusqu'à la notion de surface en y incorporant la notion de couleur. La couleur est souvent un aspect déterminant dans la phase de reconnaissance mais sa dépendance avec la luminosité nuit à sa prise en compte dans les processus de localisation et/ou de cartographie.

Si les cartes d'amers structurés à base de droites commencent à prendre une part importante dans le paysage de la cartographie, leur représentation pose un problème notamment lorsqu'il s'agit d'y associer une incertitude : anisotropie de la carte ou centrage de l'erreur sur un point sont quelques unes des difficultés liées aux représentations cartésienne ou polaire.

1.2.3 Cartes de type *hétérogène*

Bien souvent, le choix d'une représentation de l'information est induit par les mesures. Ainsi, des capteurs tels que le radar ou les nappes laser incitent une représentation polaire des points ou des droites quand une caméra ou un GPS privilégierait une représentation cartésienne. A partir de là, il n'est pas impossible d'envisager des cartes dites hétérogènes où se mêlent droites, points, plans, surface, etc. Dans cet esprit, [Folkesson 2005] propose la prise en compte au sein du même processus des amers de type droite et de type plan. Cette idée de carte diversifiée dans la structure des amers qui la composent, ne semble pas être devenue une pensée générale au sein de la littérature, qui préfère se focaliser sur la résolution des problèmes inhérents à un formalisme avant d'en introduire un autre.

1.2.4 Bilan relatif aux types de cartes

Les cartes composées de points issus de l'environnement sont encore majoritairement présentes dans les approches développées. Elles présentent l'avantage d'être souvent simples et rapides en temps de traitement. L'inconvénient majeur tient au fait qu'il faut surcharger le point d'un descripteur afin qu'il puisse être reconnu. Dans le cas du capteur choisi, à savoir la caméra, ce descripteur correspond bien souvent au voisinage du point dans l'image. En choisissant le point de tel sorte que son voisinage soit suffisamment discriminant au sein d'une zone délimitée dans l'image, on pourra le retrouver au sein de la même zone dans les images successives. Les cartes composées de droites commencent à se faire une place importante dans des environnements urbain ou intérieur. Etant moins sujettes aux bruits que les points, leur robustesse précise d'autant plus la localisation. Seules quelques applications particulières utilisent les autres types de carte et demeurent encore minoritaires dans la littérature.

1.3 Les possibilités algorithmiques

L'analyse des cartes et des capteurs va de paire avec l'analyse des algorithmes existant en matière de recherche de solution. En ce qui concerne la localisation, on peut le voir comme un problème à part entière ou comme une partie de la théorie du SLAM que nous avons déjà évoquée. Nous allons à présent aborder ces deux points en développant les différents principes de localisation existant.

1.3.1 La localisation

Localiser un objet, ou soi-même, c'est définir sa position dans un repère donné. La localisation se fait toujours par rapport à une origine et dans notre cas, ce repère sera appelé le repère monde et noté ultérieurement \mathcal{R}_m . Afin de se localiser on peut soit connaître sa position directement (localisation absolue), soit connaître sa position par rapport à un autre objet, lui-même localisé dans le repère monde (localisation relative). Etant donnés les capteurs choisis, c'est une localisation relative qui doit être développée (absence totale de capteur de positionnement absolu).

1.3.1.1 Triangulation

La question formulée par le problème de triangulation est la suivante : soit un groupe de N correspondances entre les données d'une carte et leur observation respective au sein d'une image, comment retrouver la position de l'observateur ? Ce problème illustré par la figure 1.7 a fait l'objet de nombreuses recherches et publications, et notamment en ce qui concerne les cartes de type points. On se rend bien compte que plus N est grand, plus le calcul de pose sera long mais précis. [Haralick 1994] propose une solution pour $N = 3$ points. [Quan 1999] propose à leur tour deux algorithmes pour respectivement $N = 4$ et $N = 5$ points, améliorant en précision les résultats de la méthode à 3 points sans induire un temps de

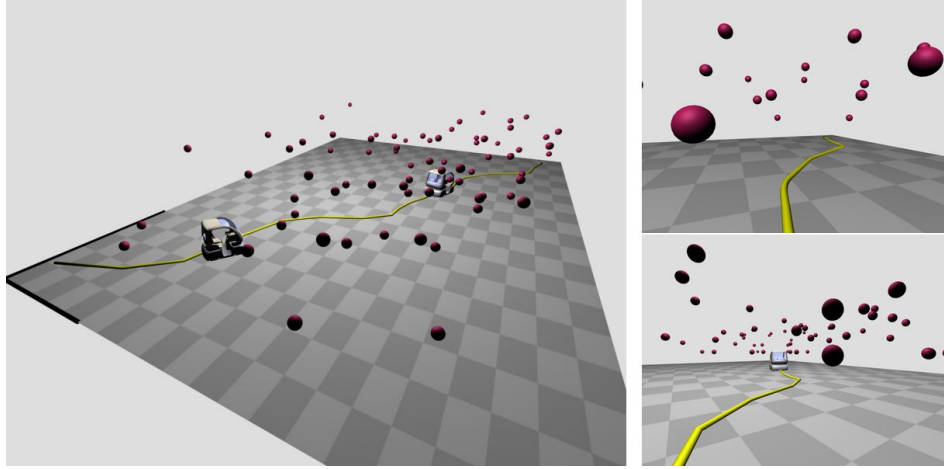


FIGURE 1.7 : **La triangulation** : soit un environnement dans lequel on sait reconnaître des amers géolocalisés (**à gauche**), comment trouver la position de chacun des véhicules en ne connaissant que l'observation qu'ils en ont (**partie droite de la figure** dans le cas d'un capteur type caméra) ?

calcul supplémentaire. Enfin, [Bujnak 2008] estime, avec $N = 4$, la pose ainsi que la focale de la caméra. La réduction du nombre de points impose que les correspondances soient parfaites et qu'aucune erreur d'association ne se soit glissée au sein des N couples. Dans la plupart des cas, les auteurs font appel à un traitement itératif reconnu sous l'acronyme de RANSAC¹ présenté par [Fischler 1981] et revisité par [Raguram 2008] pour des applications temps réel. Comme dans tous les traitements, la précision des résultats dépend d'abord et avant tout de la précision des données. Ces méthodes impliquent la connaissance des coordonnées de chaque point, ce qui sera, dans notre application, disponible dans un premier temps. Rappelons toutefois que la problématique de localisation par rapport à des amers dont la position est imprécise interviendra lorsque le véhicule sera dévié de la trajectoire de référence, pour éviter un obstacle par exemple.

1.3.1.2 Géométrie épipolaire

La géométrie épipolaire est la géométrie qui définit les contraintes qui lient plusieurs vues d'une même scène. Pour plus de simplicité dans la présentation, prenons le cas d'un point \vec{P} dans un espace 3D. L'observation de ce point depuis deux positions est donnée par la figure 1.8. Si on identifie la position/orientation de l'observateur à celle du centre optique de chaque caméra (respectivement \mathcal{O}_1 et \mathcal{O}_2), alors les trois points $(\mathcal{O}_1, \mathcal{O}_2, \vec{P})$ désignent un plan \mathcal{P} . La projection de ce plan dans chaque observation définit une droite (respectivement l et l'). Il en va de même des centres optiques $(\mathcal{O}_1, \mathcal{O}_2)$ dont les projetés appartiennent respectivement à l' et l . On définit alors la contrainte épipolaire qui énonce que les projetés d'un point dans

1. RANdom SAmple Consensus

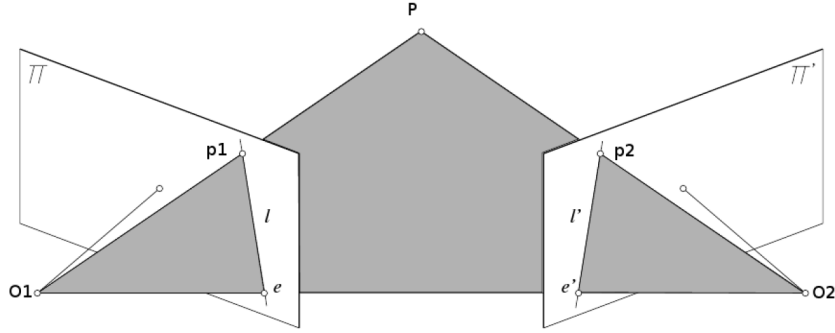


FIGURE 1.8 : Représentation d'un point vu dans 2 observations (source [Forsyth 2002]).

deux vues différentes forment un plan avec les centres optiques de chaque caméra et le point en question.

Cette contrainte est représentée par une matrice appelée *matrice essentielle* et a été introduite par [Longuet-Higgins 1981] qui proposa également une méthode d'estimation à partir de 8 points (méthode revisitée par [Hartley 1995]). La matrice essentielle contient les informations de position et d'orientation de la deuxième caméra dans le repère de la première à un facteur d'échelle près. 5 points sont théoriquement nécessaires pour calculer les paramètres de la matrice essentielle (3 pour la translation, 3 pour la rotation moins 1 pour le facteur d'échelle). Des méthodes de calcul de la matrice essentielle et d'extraction des paramètres d'orientation et de position ont été successivement proposées par [Nistér 2004][Stewenius 2006] et [Li 2006].

Généralement, ce facteur d'échelle est déterminé grâce à l'odométrie des roues. Cependant, les reliefs du sol et les phénomènes de glissement induisent rapidement des imprécisions. Remarquons que la géométrie épipolaire offre une écriture simple des projections en introduisant le facteur d'échelle au sein des paramètres. Le projeté du point \vec{P} possède non pas 2 mais 3 coordonnées (u, v, k) . La normalisation de ce vecteur de coordonnées par le facteur d'échelle k donne les coordonnées du projeté dans le plan image. Elle permet aussi une représentation à l'infini du point en posant $k = 0$. La géométrie épipolaire permet, comme la triangulation, de calculer une position relative à un ou plusieurs amers observés à condition de connaître leur position. Cela signifie que, dans le cas d'une carte inconnue, le calcul de localisation d'un amer demande une étape indépendante et supplémentaire.

1.3.1.3 L'ajustement de faisceaux

La méthode par ajustement de faisceaux donne une réponse à la principale lacune de la triangulation et de la géométrie épipolaire : elle ne nécessite pas de connaître la localisation des amers observés. La question posée est donc : comment résoudre simultanément le problème de la pose d'un observateur et des amers observés ? Elle se traduit par une erreur dans la projection des amers et le processus d'ajustement

de faisceaux tend à la réduire au minimum. Il s'agit d'un problème de minimisation globale qui tient compte en même temps de toutes les observations et de la pose de l'observateur. Cette minimisation fait appel à des algorithmes de Levenberg-Marquardt [Press 1992] ou plus simplement à des méthodes de descente de gradient.

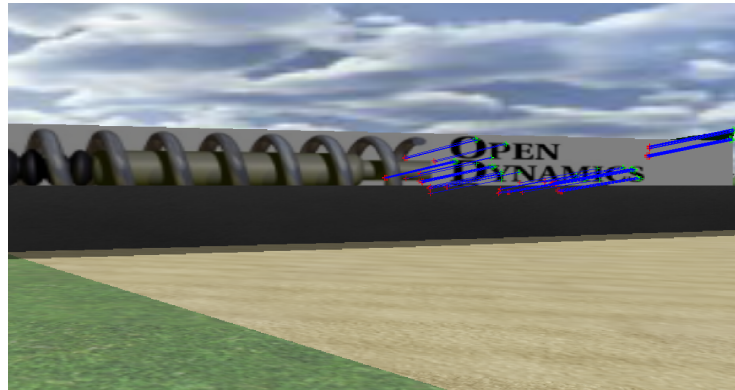
L'ajustement de faisceaux propose, compte tenu d'un certain nombre de poses de l'observateur et d'un certain nombre d'observations, de minimiser, pour chaque pose et pour chaque observation, l'erreur de reprojection. Cette erreur se caractérise, dans une image, par la somme des distances entre le projeté d'un point et son observation (voir figures 1.9). [Royer 2006] utilise cette approche pour déterminer



(a) Projection des observations (Position 1).



(b) Ajout des nouvelles observations (Position 2).



(c) Caractérisation de l'erreur de reprojection.

FIGURE 1.9 : Illustration de l'erreur de reprojection pour une carte d'amer type points et une observation par caméra. **1.9a** : depuis une position 1, on observe la projection cohérente des points de la carte (croix rouges). **1.9b** : le robot cherche à présent à connaître sa nouvelle position. Il projette donc les points de la carte dans l'image (croix rouges) et recherche les observations (croix vertes). **1.9c** : en réduisant la somme des erreurs de reprojection (lignes bleues) définies comme la distance qui sépare la projection des points de la carte (croix rouges) de leur observation (croix vertes), on trouve la nouvelle position d'observation (position 2).

une trajectoire et une carte en vue d'une application autonome. Elle nécessite un traitement long de toutes les données qui est exécuté hors ligne. Cette disponibilité de toute les données au moment du calcul est un inconvénient majeur. Une variante

a été développée par [Mouragnon 2009] pour atteindre des applications temps réel (de l'ordre de 7 images par seconde). La lenteur de ce traitement provient du grand nombre de variables impliquées dans la résolution. Dans le cadre d'une séquence de 20 vues associé à 2000 points (soit moins de 1% d'une image 640x480), le problème à résoudre est une minimisation de pas moins de 6000 variables. C'est justement cette quantité de variables que [Konolige 2008] proposent de réduire pour tendre vers un ajustement de faisceaux temps réel. Bien que la littérature semble parfaitement répondre à notre problème de localisation, la disponibilité de l'intégralité des données et le temps de calcul nécessaire ne jouent pas en notre faveur. Remarquons enfin qu'aucune information d'incertitude n'est fournie par ces méthodes et l'estimation de celle-ci doit se faire par un traitement supplémentaire.

1.3.1.4 Filtre de Kalman

L'estimation bayésienne s'appuie sur l'estimation de la densité de probabilité d'une variable aléatoire souvent considérée comme gaussienne. En se plaçant dans le cadre des fonctions markoviennes d'ordre 1, le filtre de Kalman permet l'estimation simultanée de paramètres et de l'incertitude associée au travers de la matrice de variance-covariance avec une hypothèse gaussienne. D'une manière générale, en estimation, on cherche la solution qui soit, diffère le moins, soit, ressemble le plus. Cette réflexion quelque peu naïve permet de faire le parallèle avec les estimateurs de minimum de variance et de maximum de vraisemblance qui ne sont optimaux que dans certains cas (dont le cas gaussien). Dans le cas des fonctions linéaires, [Ristic 2004] montre que le filtre de Kalman atteint la borne de Cramer-Rao qui est le minimum de variance d'un estimateur (condition d'optimalité). Par opposition et pour clore la comparaison, la méthode par ajustement de faisceaux s'apparente au calcul d'un maximum de vraisemblance.

Le filtre de Kalman se décompose en 3 étapes principales : la prédiction, le calcul du gain et la mise à jour. Soit un vecteur de paramètres \vec{X} à estimer associé à une matrice d'incertitude \mathbf{P} , la phase de prédiction, souvent bâtie sur un modèle d'évolution en robotique mobile, permet d'estimer dans un premier temps le vecteur d'état et son incertitude. Chaque observation vient alors alimenter le calcul du gain en tenant compte à chaque fois des incertitudes respectives. Finalement, l'étape de mise à jour vient corriger les paramètres du vecteur d'état et préciser l'incertitude de chacun d'eux. Cette étape détermine aussi les covariances, c'est-à-dire le lien statistique qui existe entre les paramètres. Ces trois étapes sont souvent réduites à deux en regroupant le calcul du gain et la mise à jour [Mourllion 2006] :

– **Prédiction** : calcul de l'état *a priori*

$$\vec{X}_{k+1|k} = \mathbf{f}(\vec{X}_{k|k}, u_{k+1}) \quad (1.1)$$

$$\mathbf{P}_{k+1|k} = \mathbf{F}_{k+1} \mathbf{P}_{k|k} \mathbf{F}_{k+1}^T + \mathbf{Q}_{k+1} \quad (1.2)$$

où \mathbf{F}_{k+1} est la jacobienne de la fonction d'évolution $f(\vec{X}_{k|k}, u_{k+1})$ qui tient compte de l'état précédent et des entrées u_{k+1} du système. \mathbf{Q}_{k+1} est la matrice de covariance du bruit d'évolution du processus.

– **Mise à jour** : calcul de l'état *a posteriori*

$$\mathbf{G}_{k+1} = \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T + \mathbf{R}\mathbf{o}_{k+1})^{-1} \quad (1.3)$$

$$\vec{X}_{k+1|k+1} = \vec{X}_{k+1|k} + \mathbf{G}_{k+1} (\mathbf{z}_{k+1} - \mathbf{h}(\vec{X}_{k+1|k})) \quad (1.4)$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{G}_{k+1} \mathbf{H}_{k+1} \mathbf{P}_{k+1|k} \quad (1.5)$$

où \mathbf{H}_{k+1} est la jacobienne de la fonction d'observation $\mathbf{h}(\vec{X}_{k+1|k})$. $\mathbf{R}\mathbf{o}_{k+1}$ est la matrice de variance-covariance du bruit d'observation et \mathbf{G}_{k+1} désigne le gain de Kalman. Le cas des fonctions d'évolution et d'observation non-linéaires a fait l'objet d'une variante : le filtre de Kalman étendu ou EKF (Extended Kalman Filter).

1.3.1.5 le filtre de Kalman étendu ou EKF

Le filtre de Kalman est optimal pour des fonctions linéaires dans le cas gaussien et le reste pour des fonctions non-linéaires à condition de linéariser autour de la valeur vraie du vecteur de paramètres \vec{X} . Cette valeur vraie est souvent indisponible puisqu'il s'agit précisément de la valeur cherchée au travers de son estimation $\vec{X}_{k|k}$. Par conséquent, la linéarisation est faite autour de l'estimée $\vec{X}_{k|k}$. Les équations du filtre de Kalman restent inchangées si ce n'est que les matrices \mathbf{F}_{k+1} et \mathbf{H}_{k+1} traduisent respectivement la linéarisation des fonctions d'évolution et d'observation. Dans le cas d'un développement de Taylor à l'ordre 1, les linéarisations sont égales aux jacobienes et le filtre de Kalman est linéaire. Ces dernières années, ce filtre a été très largement étudié dans différents domaines comme la robotique, la météorologie [Palmer 2006] ou encore la reconnaissance faciale [Maghami 2007]. Dans le domaine de la robotique mobile, citons les travaux de [Bar-Shalom 1988] qui donne un exemple de réalisation de localisation dans un monde 2D. Cependant, [Castellanos 2004] a montré que le filtre de Kalman était inconsistant, c'est-à-dire que les non-linéarités quelles qu'elles soient finissent toujours par engendrer une dérive qui rend l'estimation non intègre (la vraie valeur n'appartient plus au domaine d'incertitude de l'estimation). Dans le cas d'une recherche de position absolue, cette inconsistance reste problématique.

1.3.1.6 UKF

Le filtre de Kalman est sous-optimal si les fonctions d'évolution ou d'observation sont non-linéaires (car linéarisée autour de l'estimée et non autour de la valeur vraie). La variante UKF (Unscented Kalman Filter) propose une méthode pour calculer les moments d'ordre 1 et 2 de la phase *a posteriori*. Des états pondérés par une fonction de vraisemblance sont créés autour de l'espérance et sont propagés au travers de la fonction non-linéaire. Ces états, appelés particules, sont ré-échantillonnés par la suite et soumis aux estimateurs de moyenne et de variance au terme d'un nombre défini d'itérations. Les résultats obtenus sont plus pertinents qu'avec un EKF mais le temps de calcul supplémentaire rend bien souvent le traitement irréalisable en temps réel. Le principal avantage de cette méthode est de se passer du calcul des

jacobiennes où de toute linéarisation impropre d'une fonction non-linéaire au sein du filtre de Kalman. Les travaux de [Julier 2000], [Julier 2004] et [Wan 2000] posent les bases de l'utilisation de cette variante dans le cadre de la robotique mobile. La notion de particules dans l'estimation des moments statistiques *a posteriori* fait appel à la théorie des filtres particulaires et [Chen 2009] en donne un exemple récent. On notera pour finir que la création de particules autour d'une estimation (méthode de Monte-Carlo), permet de gérer des transitions non-linéaires entre deux espaces d'incertitude.

1.3.1.7 Filtre d'information

Une dernière variante du filtre de Kalman est sans doute le filtre d'information [Gibbens 2000]. Il définit la matrice d'information comme étant l'inverse de la matrice de covariance du vecteur d'état estimé. Le vecteur d'information associé est lié à l'état par la relation suivante :

$$\begin{aligned} \mathbf{Y}_{k+1|k+1} &= \mathbf{P}_{k+1|k+1}^{-1} \\ \vec{y}_{k+1|k+1} &= \mathbf{P}_{k+1|k+1}^{-1} \vec{X}_{k+1|k+1} \end{aligned} \quad (1.6)$$

Son principal avantage est de pouvoir considérer indépendamment les mesures par simple addition de leurs effets au vecteur d'information. [Mahon 2008] développe l'application d'un tel filtre, localisant simultanément le véhicule et les amers observés. L'inconvénient réside principalement dans l'utilisation ultérieure du résultat. En effet, pour retrouver le vecteur d'état résultant, une conversion est nécessaire. Cette opération, de même que pour chaque mesure, repose principalement sur une inversion de matrice et donc coûteux en temps de traitement.

1.3.1.8 Une méthode globale ou une méthode itérative ?

Cette question, bien souvent en marge des algorithmes, est soulevée par le filtre d'information et par l'interdépendance des observations. Pour illustrer la question prenons un cas simple de recherche de visage dans une photo. On sait qu'un visage est composé de deux yeux. La méthode globale va chercher chaque oeil de manière indépendante tandis que la méthode itérative va s'appuyer sur le résultat des observations précédentes. Ainsi, c'est dans la différence de surface de la zone de recherche que réside l'accélération de la version itérative. Son inconvénient principal se situe au niveau des mauvaises associations de données. La figure 1.10 illustre le cas d'échec de la méthode itérative par rapport à la méthode globale. Pour développer un peu ce cas et voir les améliorations effectuées dans le domaine, prenons l'exemple simple de la recherche des paramètres d'une équation de parabole. Cette courbe est définie par trois coefficients (a , b , c) liés par la relation suivante :

$$ax^2 + bx + c = y \quad (1.7)$$

où (x , y) sont les coordonnées d'un point de la parabole. Le figure 1.11 illustre le cas (a , b , c) = (1, 2, 3) qui sera la solution recherchée à partir d'observations de points



(a) Deux yeux et une bouche pour un visage (b) Comparaison des résultats obtenus par une méthode globale et une méthode itérative

FIGURE 1.10 : Supposons qu'un visage se reconnaît par un œil gauche, œil droit et une bouche, les deux méthodes donneraient le même résultat, à savoir le bon, dans le cas de la figure 1.10a. Si on suppose que la méthode itérative commence par rechercher l'œil gauche, alors la figure 1.10b montre que la méthode globale trouve le bon résultat (ellipse verte) tandis que pour l'œil gauche trouvé (carré vert), l'algorithme itératif cherchera l'œil droit et la bouche au mauvais endroit (respectivement dans les zones en pointillées bleu et orange). Le résultat de cette dernière méthode (pointillé rouge) est incohérent et montre les précautions à prendre dans l'utilisation d'un algorithme itératif.

pris aléatoirement sur la courbe et bruités par un bruit blanc gaussien. La méthode

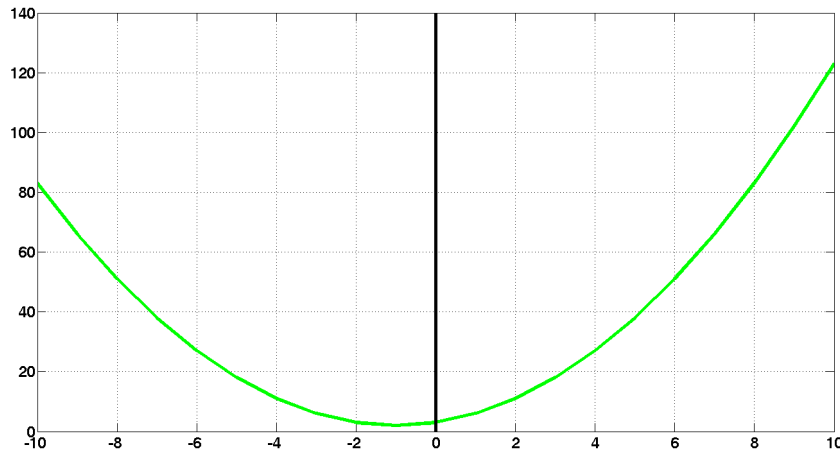


FIGURE 1.11 : Parabole d'équation $y = 1x^2 + 2x + 3$.

globale, disposant de toutes les observations en amont du traitement, peut extraire les points aberrants (avec un algorithme de RANSAC par exemple). Cette approche calcule un moindres carrés avec les observations restantes (reconnues comme cohérentes par l'algorithme de RANSAC). La méthode itérative ne dispose pas de toutes les observations et par conséquent, elle n'a pas de moyen de déterminer quelles sont les données aberrantes s'il y en a. Une méthode serait de prendre les observations dont l'incertitude se trouve à une tolérance donnée par rapport à l'estimation. Cependant,

lorsqu'il s'agit des premiers points, cette tolérance est suffisamment importante pour ne pas les rejeter. Le tableau 1.2 donne des résultats de comparaison sur 1000 expériences de 100 points pris aléatoirement sur la courbe pour chaque méthode. Il

Méthode	écart moyen	écart max	écart min
Itérative	2.9173	9.2922	0.10818
Globale	0.16642	0.6194	0.14296

TABLE 1.2 : résultat d'estimation des paramètres de la parabole sur 1000 expériences de 100 valeurs.

montre que la version itérative donne de moins bons résultats que la version globale. Cependant, [Malartre 2011] montre, dans le cas d'un algorithme EKF, que la complexité est plus importante dans le cas de la méthode globale (voir le tableau 1.3). La contre-performance de la méthode itérative par rapport à la méthode globale est

Méthode	Complexité en fonction de N points dans la carte
Globale	$\mathcal{O}(340N^3 + 612N^2 + 576N + 144)$
Itérative	$\mathcal{O}(80N^3 + 558N^2 + 1024N)$

TABLE 1.3 : Complexité du filtre de Kalman étendu.

due aux premiers points de l'échantillon qui s'avèrent être loin de la courbe cherchée. En effet, la méthode globale disposant de toutes les observations, il lui est possible d'éliminer celles détectées comme aberrantes. Dans le cas de la méthode itérative, les résultats étant issus sur la mise à jour induite par ces points fortement erronés, l'algorithme ne peut « redresser son comportement ». L'algorithme JCBB (Joint Compatibility Branch and Bound) développé par [Neira 2001] propose une manière de sélectionner un ensemble de paires prédiction-observation en rejetant les appariements détectés comme faux. Ce traitement, gourmand en taille mémoire, présente l'avantage de pouvoir traiter rapidement le problème de l'association de données au fur et à mesure que les observations et les mises à jour sont effectuées. Pour cela, le processus s'autorise de revenir en arrière lorsqu'une voie explorée est jugée incompatible avec les observations futures. Sur un autre principe, [Civera 2009] propose de calculer le point de linéarisation des fonctions à partir de la mise à jour d'un point sélectionné par un algorithme de RANSAC. Cette solution implique toujours de posséder au moment du calcul l'intégralité des observations. Tant que la phase de recherche et d'extraction des mesures n'est pas prépondérante dans le traitement de l'information, cette dernière solution est acceptable et fournit des résultats équivalents à la méthode globale pour un gain de temps significatif. Dans le cas contraire, la méthode itérative reste désignée, à condition d'optimiser la recherche de toutes les observations.

1.3.2 Le SLAM

Le processus SLAM (Simultaneous Localization And Mapping) [Leonard 1991] a déjà largement été évoqué au sein des références bibliographiques ainsi que dans le début de ce manuscrit et pour cause : il est bien rare que l'environnement soit connu avant le déroulement d'une mission. Même si notre fonction première de localisation avec une carte de référence connue ne traduit que la moitié de la problématique du SLAM, bon nombre des améliorations des algorithmes sont inscrites dans ce cadre. D'un point de vue théorique, il a été montré que le problème du SLAM converge [Csorba 1998] [Cole 2006] à condition de fermer des boucles au cours du parcours (cela consiste en la ré-observation d'informations de l'environnement déjà cartographié). Dans des applications monoculaires, [Davison 2007] et [Mouragnon 2009] ont apporté des avancées significatives en adaptant respectivement une approche bayésienne et un ajustement de faisceau dans l'estimation simultanée de la position du véhicule et d'amers points. Comme développé plus haut les approches diffèrent principalement dans le type de carte et dans les algorithmes, minimisant la variance ou maximisant la vraisemblance. [Thrun 2000] propose de mêler un filtre de Kalman et une approche particulière, à l'image de l'UKF, donnant des résultats de localisation significatifs par rapport à une vérité terrain.

1.3.3 Bilan

Les approches dites globales ou locales séparent les algorithmes en deux philosophies aux problématiques différentes. Les méthodes locales approximent les équations d'état ou d'observation par leur développement de Taylor au premier et plus rarement au deuxième ordre. Elles fournissent un résultat qui tient compte d'une faible partie de l'espace d'état. Les méthodes qui fournissent des résultats valides avec presque l'intégralité de l'espace d'état sont appelées globales par opposition. Citons [Straka 2011] qui compare les temps de traitement et les erreurs euclidiennes du résultat par rapport à la vérité terrain pour des algorithmes globaux et locaux. Il constate que la précision des algorithmes globaux est dégradée par des temps de traitement en dehors du cadre temps réel et inversement pour les algorithmes locaux (qui donnent des résultats moins précis pour un coût calculatoire plus faible). Du point de vue de la contrainte temps réel, le filtre de Kalman et sa version étendue (EKF) s'avère être une bonne solution. Les problèmes de non-linéarité ont ouvert une voie dans laquelle l'UKF donne de meilleurs résultats. Cependant, encore une fois, l'estimation de l'incertitude a posteriori implique bien souvent un filtre particulier qui ne permet plus des applications temps réel. Le filtre d'information offre a priori une optimisation en temps de traitement mais son utilisation implique une conversion à chaque étape des paramètres afin qu'ils puissent être utilisés par les modules de commande. Cette conversion engendrerait un surplus de calcul supérieur à l'optimisation de la complexité. Finalement, tout n'est qu'une question de position et de contraintes : le choix doit se faire en regard des obligations, des résultats attendus, du cadre du projet et fera l'objet de la section suivante.

1.4 Choix et conséquences

Au terme de cette analyse, il faut faire deux choix : le choix du type de carte et le choix de l'approche de localisation, chacun d'eux amenant son lot de contraintes qu'il faut mêler à celles du cadre des travaux de thèse. De plus, ces choix impliquent bien souvent des approximations engendrant des erreurs ou des biais non désirés.

1.4.1 Choix de la carte

Les contraintes de rejeu sous une variation de luminosité importante ne sont pas en faveur des cartes structurées de type surface. D'une manière générale, les cadences temps réel de traitement ne sont généralement pas à la faveur des amers structurés et c'est principalement ce qui a guidé notre choix vers les cartes de type points.

La séduisante simplicité de la paramétrisation cartésienne pose le problème de la représentation des points à l'infini tandis que le problème est déplacé sur des points proches avec la représentation en inverse de la profondeur. La représentation sphérique, quoique plus proche du mouvement humain, ne permet pas une écriture simple des translations et n'a pas été retenue. Finalement, pour des raisons de non-observabilité des paramètres de la représentation en inverse et en prenant le parti de résoudre la configuration défailante, nous avons choisi de représenter les points de la carte de manière cartésienne.

Ce choix implique de choisir un descripteur associé à ce point afin de pouvoir le reconnaître par la suite. Il est courant dans des applications visuelles, de prendre le voisinage du point dans l'image, mais de quelle taille ? [Davison 1998] et [Davison 2003] montrent que pour des applications impliquant l'observation d'un même amer sur un grand nombre d'images, il faut un descripteur large. En suivant cette proposition, nous avons choisi une taille de 15 lignes par 15 colonnes (225 pixels).

Le projet souhaite travailler avec des cartes légères et facilement transportables. On mentionne ici une taille de 10 Mo par kilomètre uniquement pour la carte. La trajectoire sera stockée d'une autre manière et contiendra de toute façon la même quantité d'information quel que soit le codage des données de la carte choisi. Des analyses en terme de quantité d'information ont été menées dans [Féraud 2010a] et un formalisme de carte a été choisi. Chaque point sera défini par 3 coordonnées cartésiennes (sur 32 bits chacune), par les variances sur chaque axes du repère monde (aussi sur 32 bits) et par la position de première observation (encore une fois, 3 coordonnées définies sur 32 bits chacune). A cela n'oublions pas le descripteur dont chaque pixel sera codé sur 8 bits. Au final, nous arrivons à la quantité de données suivante pour N points :

$$Qd = (32 \times (3 + 3 + 3) + 8 \times (15 \times 15)) \times N \quad (1.8)$$

Pour 10 Mo cela représente environ 4780 points ce qui est faible pour ce type d'application. Nous verrons plus tard que nous pouvons construire une carte avec

suffisamment d'information, tout en respectant cette contrainte, qui permet une localisation précise à 10 cm d'une trajectoire de référence. Par ailleurs, nous montrerons des résultats sur données réelles avec une carte de l'ordre de 5 Mo pour un kilomètre (par conséquent, un nombre plus important de points améliorera les résultats de localisation).

1.4.2 Choix de l'approche

La connaissance de l'incertitude associée aux paramètres de la pose du véhicule est nécessaire pour qualifier le résultat de la fonction de localisation. Son estimation de manière simultanée au sein de l'algorithme nous a fait privilégier le filtre de Kalman. L'optique temps réel est une contrainte forte du cadre de la thèse et c'est pourquoi c'est une version EKF plutôt que UKF qui a été pressentie dans le cas de fonctions non-linéaires.

Si on se réfère à l'aspect temps réel de la fonction, une version itérative de l'algorithme apparaît comme le meilleur choix. Cette approche, aussi identifiée comme approche focalisante, doit prendre en compte l'erreur d'association de données afin de minimiser les cas de divergence. Par ailleurs, l'algorithme EKF se prête bien à la fusion de données ainsi qu'à l'ajout de nouvelles informations (comme la connaissance d'un véhicule leader observé depuis un véhicule suiveur et reconnu à partir d'association avec une banque de données par exemple).

1.4.3 Verrou technologique

Les choix faits précédemment ne sont pas sans conséquences et présentent des verrous scientifiques importants. L'expression de la non-linéarité d'une fonction d'observation, par exemple, se traduit à deux niveaux : dans la définition d'une zone de recherche et dans la mise à jour du gain de Kalman. Des solutions seront apportées ultérieurement dans chacun des cas (voir chapitre 2 et 3). De plus, le filtre de Kalman impose de travailler avec des modèles de bruits gaussiens qui ne reflètent pas exactement la réalité. Ces écarts entre modélisation et réalité se traduisent lors de l'initialisation des points de la carte. Dans un premier temps, pour une localisation sur une carte de référence pré-établie, cela n'est pas un problème, mais cela le sera plus dans le deuxième volet de la problématique où il est question de réaliser, à proprement parler, un SLAM hybridant carte propre et carte de référence.

Localisation

Sommaire

2.1	Introduction	34
2.2	Rappels sur le filtre de Kalman	34
2.3	Localisation en ligne	35
2.3.1	Prédiction de l'état du véhicule	35
2.3.2	Prédiction de l'observation	40
2.3.3	Recherche de l'observation - Association	49
2.3.4	Mise à jour	59
2.3.5	Résultats	61
2.4	Localisation initiale	62
2.4.1	Problème de l'initialisation	62
2.4.2	Solution envisagée	63
2.4.3	Résultats de localisation initiale	65

2.1 Introduction

La localisation du véhicule est le problème principal abordé dans cette partie. Il s'agit de localiser un véhicule sur un chemin avec une précision suffisante (inférieure à 10 cm), en connaissant une carte des amers points observables ainsi que la trajectoire à suivre.

Dans le chapitre précédent, nous avons argumenté nos choix quant à la nature de la carte et aux différents algorithmes. Le but étant, dans un premier temps, de construire une fonction de localisation d'un véhicule à l'aide d'une carte. Suite à cette localisation, on souhaite faire rejouer une trajectoire à un véhicule de manière automatique avec une erreur inférieure à 10 cm par rapport à une trajectoire donnée. Dans le cas d'un convoi, par exemple, la carte pourrait être construite par le premier véhicule et transmise aux suivants qui doivent évoluer au plus près du chemin suivi par le premier véhicule. Il ne s'agit donc pas ici d'étudier comment le premier passage construit la carte, mais bien quelle utilisation peut en faire un autre véhicule et quels résultats de localisation peuvent être attendus.

Dans le cadre du Visual SLAM (SLAM intégrant un capteur visuel), nous avons fait le choix d'une carte composée de points 3D avec, pour chacun, un descripteur identifié par son voisinage dans l'image (15x15 pixels). De plus, la position de sa première observation est elle aussi enregistrée afin de ne pas avoir à étudier toute la carte à chaque instant, mais seulement une partie correspondant aux points supposés observables. Enfin, les données proprioceptives du véhicule sont disponibles à chaque instant. De ce fait, le problème devient alors le suivant : soit une observation de l'environnement par caméra à un instant donné, quelle est la position du véhicule correspondante ?

Ce problème se décompose presque naturellement en deux cas de figure. Premièrement, partons du cas où une connaissance a priori est disponible, comme cela peut l'être au cours du traitement de la mission. Cet a priori se traduit par une incertitude de localisation faible de l'ordre du degré pour les angles d'orientation et de la dizaine de centimètre pour la position. Dans cette configuration, il faudra mettre à jour la position et l'orientation du véhicule en prenant en compte l'incertitude de toutes les informations. Deuxièmement, le cas de la première localisation, sans connaissance a priori, reste un problème complexe. N'ayant qu'une approximation grossière de la position, les possibilités de mise en correspondance entre une observation et les points de la carte de référence peuvent être multiples. Chacun de ces deux cas vont être étudiés successivement dans les sections 2.3 et 2.4 et des résultats de simulations seront apportés.

2.2 Rappels sur le filtre de Kalman

En préambule de ce qui va suivre, rappelons les relations du filtre de Kalman étendu ainsi que la signification de chaque élément. Le filtre de Kalman se décompose en 3 étapes qui sont la prédiction, le calcul du gain de correction et la mise à jour.

Les relations sont les suivantes :

$$\text{Prédiction} \quad \begin{cases} \vec{X}_{k+1|k} = \mathbf{f}(\vec{X}_{k|k}, u_{k+1}) \\ \mathbf{P}_{k+1|k} = \mathbf{F}_{k+1} \mathbf{P}_{k|k} \mathbf{F}_{k+1}^T + \mathbf{Q}_{k+1} \end{cases} \quad (2.1)$$

$$\text{Gain} \quad \begin{cases} \mathbf{G}_{k+1} = \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T (\mathbf{H}_{k+1} \mathbf{P}_{k+1|k} \mathbf{H}_{k+1}^T + \mathbf{R}_{o_{k+1}})^{-1} \end{cases} \quad (2.2)$$

$$\text{Mise à jour} \quad \begin{cases} \vec{X}_{k+1|k+1} = \vec{X}_{k+1|k} + \mathbf{G}_{k+1} (\mathbf{z}_{k+1} - \mathbf{h}(\vec{X}_{k+1|k})) \\ \mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{G}_{k+1} \mathbf{H}_{k+1} \mathbf{P}_{k+1|k} \end{cases} \quad (2.3)$$

$\vec{X}_{k|k}$ est le vecteur d'état, c'est-à-dire l'ensemble des paramètres que l'on souhaite faire évoluer et venir préciser à l'aide du filtre. Dans cette première approche de localisation, la carte étant connue, seuls les paramètres du véhicule seront à estimer et constitueront le vecteur d'état. $\mathbf{P}_{k|k}$ est la matrice d'incertitude associée au vecteur d'état $\vec{X}_{k|k}$. Les matrices \mathbf{Q} et \mathbf{R}_o sont respectivement les matrices du bruit de processus et du bruit de mesure. Elles permettent d'autoriser des libertés d'évolution plus ou moins importantes. Elles sont nécessaires étant donné que les systèmes et les modèles sont bien souvent hypostatiques face aux nombres de paramètres du vecteur d'état. Nous en arrivons enfin au modèle et nous rappelons que les fonctions \mathbf{f} et \mathbf{h} sont respectivement la fonction d'évolution et la fonction d'observation. Chacune doit être définie et leur jacobienne par rapport au vecteur d'état, respectivement \mathbf{F}_k et \mathbf{H}_k doivent être calculées. Bien définir et bien poser les équations et relations d'observation et de prédiction sont bien souvent la clé d'un système convergent. Le dimensionnement des différents bruits contraint l'intégrité de l'estimation vis à vis des paramètres recherchés.

2.3 Localisation en ligne

La localisation en ligne est le processus permettant d'estimer la pose d'un véhicule à l'instant $k + 1$ à partir d'une connaissance précise de sa position et de son orientation à l'instant k , compte tenu des données issues des différents capteurs.

2.3.1 Prédiction de l'état du véhicule

Comme rappelé dans la section 1.3.1.4 et précédemment, nous utiliserons un filtre de Kalman dont la première étape consiste en une étape de prédiction. Dans le cas d'un capteur à fréquence élevée, la méthode la plus courante est de faire l'étape de prédiction en utilisant les informations de ce capteur. Dans le cas contraire, il doit alors être pris comme une observation et par conséquent, un modèle à vitesse constante ou accélération constante est le plus souvent utilisé pour résoudre l'étape de prédiction. Dans notre cas, les informations proprioceptives sont disponibles à une fréquence suffisamment élevée par rapport à la vitesse du véhicule pour qu'on puisse les intégrer dans le modèle de prédiction. Le modèle d'évolution du véhicule est supposé ici assimilé au modèle bicyclette, ce qui est courant dans ce genre d'application mobile (illustré dans la figure 2.1). Le véhicule possède 6 paramètres de

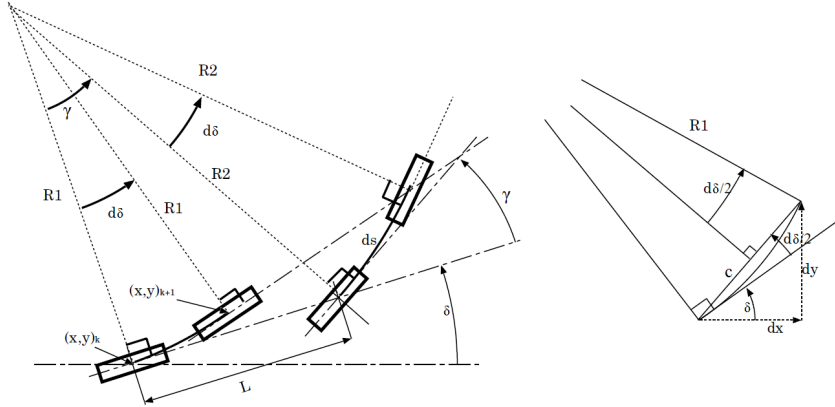


FIGURE 2.1 : Représentation du modèle d'évolution bicyclette.

localisation à savoir : 3 paramètres de translation (X_v Y_v Z_v) et 3 paramètres définissant les angles de rotation (α_x α_y α_z), respectivement autour des axes \vec{x} , \vec{y} et \vec{z} . La figure 2.2 rappelle la représentation du repère véhicule \mathcal{R}_v dans le repère monde. De plus, les faibles variations de vitesse permettent de mettre en place les

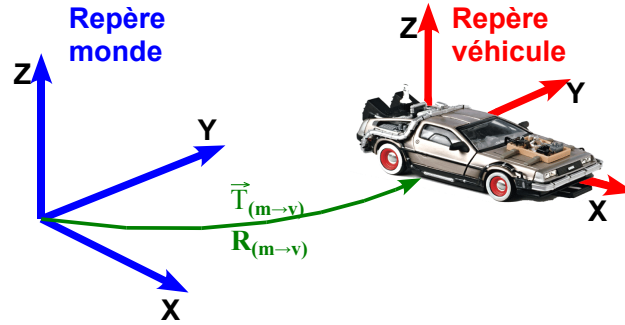


FIGURE 2.2 : Représentation du repère véhicule dans le repère monde. Le passage du repère monde au repère véhicule se fait au travers du vecteur de translation $\vec{T}_{(m \rightarrow v)}$ et de la matrice de rotation $\mathbf{R}_{(m \rightarrow v)}$.

équations du modèle à vitesse constante et à angles constants.

$$\begin{aligned}
 \gamma &= \frac{ds \sin(\delta)}{L} \quad \text{et} \quad c = \frac{2L}{\tan(\delta)} \sin\left(\frac{\gamma}{2}\right) \\
 X_{v_{k+1}} &= X_{v_k} + c \times \cos\left(\alpha_{z_k} + \frac{\gamma}{2}\right) \times \cos(\alpha_{y_k}) \\
 Y_{v_{k+1}} &= Y_{v_k} + c \times \sin\left(\alpha_{z_k} + \frac{\gamma}{2}\right) \times \cos(\alpha_{y_k}) \\
 Z_{v_{k+1}} &= Z_{v_k} - c \times \sin(\alpha_{y_k}) \\
 \alpha_{x_{k+1}} &= \alpha_{x_k} \\
 \alpha_{y_{k+1}} &= \alpha_{y_k} \\
 \alpha_{z_{k+1}} &= \alpha_{z_k} + \gamma
 \end{aligned} \tag{2.4}$$

où ds est la valeur donnée par l'odomètre (soit l'abscisse curviligne parcourue par la roue directrice) et δ est l'angle que fait la roue directrice avec le reste du véhicule. L est l'empattement du véhicule. Ce modèle présente une singularité lorsque l'angle volant δ est nul. En effet, l'apparition d'une division par zéro impose une étude asymptotique du modèle. On effectue un développement limité en zéro :

$$\begin{aligned}
 \delta \rightarrow 0 &\Rightarrow \gamma \rightarrow 0 \text{ et } c \rightarrow ds \\
 X_{v_{k+1}} &\rightarrow X_{v_k} + c \times \cos(\alpha_{z_k}) \times \cos(\alpha_{y_k}) \\
 Y_{v_{k+1}} &\rightarrow Y_{v_k} + c \times \sin(\alpha_{z_k}) \times \cos(\alpha_{y_k}) \\
 Z_{v_{k+1}} &\rightarrow Z_{v_k} - c \times \sin(\alpha_{y_k}) \\
 \alpha_{x_{k+1}} &\rightarrow \alpha_{x_k} \\
 \alpha_{y_{k+1}} &\rightarrow \alpha_{y_k} \\
 \alpha_{z_{k+1}} &\rightarrow \alpha_{z_k}
 \end{aligned} \tag{2.5}$$

A ce stade, il est possible de reconstruire un trajet bidimensionnel du véhicule. En effet, en ayant réalisé une expérimentation durant laquelle sont enregistrées les informations proprioceptives, il est tout à fait possible de retracer une approximation du chemin suivi uniquement à l'aide du modèle d'évolution. A partir d'un jeu de données de simulation, la figure 2.3 illustre le propos.

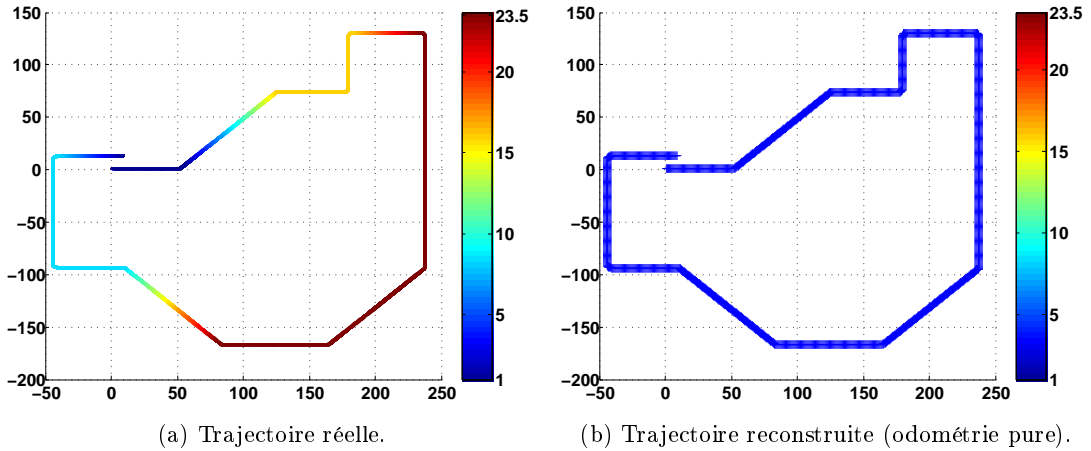


FIGURE 2.3 : **2.3a** : La trajectoire de simulation réellement exécutée. **2.3b** : La trajectoire reconstruite à partir des données proprioceptives. On constate que les variations d'altitude (couleur de la courbe en référence à l'échelle fournie) ne sont pas estimées par le modèle d'évolution. Les données proprioceptives sont supposées parfaites.

Dans le cadre du filtre de Kalman, l'incertitude, au travers de la matrice de variance-covariance, est estimée. Suite à cette étape d'évolution de l'état du véhicule, la covariance doit elle aussi être ré-évaluée. Pour ce faire, il faut calculer les jacobiniennes du modèle d'évolution selon les variables d'état (X_{v_k} , Y_{v_k} , Z_{v_k} , α_{x_k} ,

$\alpha_{y_k}, \alpha_{z_k}$) et selon les paramètres d'entrée δ et ds . Les relations 2.6 donnent les éléments constants de la jacobienne tandis que celles données par 2.7, 2.8, 2.9 et 2.10 fournissent les équations des éléments variables.

$$\begin{aligned}
1 &= \frac{\partial X_{v_{k+1}}}{\partial X_{v_k}} = \frac{\partial Y_{v_{k+1}}}{\partial Y_{v_k}} = \frac{\partial Z_{v_{k+1}}}{\partial Z_{v_k}} = \frac{\partial \alpha_{x_k}}{\partial \alpha_{x_k}} = \frac{\partial \alpha_{y_{k+1}}}{\partial \alpha_{y_k}} = \frac{\partial \alpha_{z_{k+1}}}{\partial \alpha_{z_k}} \\
0 &= \frac{\partial X_{v_{k+1}}}{\partial Y_{v_k}} = \frac{\partial X_{v_{k+1}}}{\partial Z_{v_k}} = \frac{\partial X_{v_{k+1}}}{\partial \alpha_{x_k}} \\
0 &= \frac{\partial Y_{v_{k+1}}}{\partial X_{v_k}} = \frac{\partial Y_{v_{k+1}}}{\partial Z_{v_k}} = \frac{\partial Y_{v_{k+1}}}{\partial \alpha_{x_k}} \\
0 &= \frac{\partial Z_{v_{k+1}}}{\partial X_{v_k}} = \frac{\partial Z_{v_{k+1}}}{\partial Y_{v_k}} = \frac{\partial Z_{v_{k+1}}}{\partial \alpha_{x_k}} = \frac{\partial Z_{v_{k+1}}}{\partial \alpha_{z_k}} \\
0 &= \frac{\partial \alpha_{x_{k+1}}}{\partial X_{v_k}} = \frac{\partial \alpha_{x_{k+1}}}{\partial Y_{v_k}} = \frac{\partial \alpha_{x_{k+1}}}{\partial Z_{v_k}} = \frac{\partial \alpha_{x_{k+1}}}{\partial \alpha_{y_k}} = \frac{\partial \alpha_{x_{k+1}}}{\partial \alpha_{z_k}} \\
0 &= \frac{\partial \alpha_{y_{k+1}}}{\partial X_{v_k}} = \frac{\partial \alpha_{y_{k+1}}}{\partial Y_{v_k}} = \frac{\partial \alpha_{y_{k+1}}}{\partial Z_{v_k}} = \frac{\partial \alpha_{y_{k+1}}}{\partial \alpha_{x_k}} = \frac{\partial \alpha_{y_{k+1}}}{\partial \alpha_{z_k}} \\
0 &= \frac{\partial \alpha_{z_{k+1}}}{\partial X_{v_k}} = \frac{\partial \alpha_{z_{k+1}}}{\partial Y_{v_k}} = \frac{\partial \alpha_{z_{k+1}}}{\partial Z_{v_k}} = \frac{\partial \alpha_{z_{k+1}}}{\partial \alpha_{x_k}} = \frac{\partial \alpha_{z_{k+1}}}{\partial \alpha_{y_k}}
\end{aligned} \tag{2.6}$$

$$\begin{aligned}
\gamma &= \frac{ds \sin(\delta)}{L} \quad \text{et} \quad c = \frac{2L}{\tan(\delta)} \sin\left(\frac{\gamma}{2}\right) \\
\frac{\partial X_{v_{k+1}}}{\partial \alpha_{y_k}} &= -c \times \cos\left(\alpha_{z_k} + \frac{\gamma}{2}\right) \times \sin(\alpha_{y_k}) \\
\frac{\partial X_{v_{k+1}}}{\partial \alpha_{z_k}} &= -c \times \sin\left(\alpha_{z_k} + \frac{\gamma}{2}\right) \times \cos(\alpha_{y_k}) \\
\frac{\partial X_{v_{k+1}}}{\partial ds} &= \cos(\delta) \times \cos(\alpha_{z_k} + \gamma) \times \cos(\alpha_{y_k}) \\
\frac{\partial X_{v_{k+1}}}{\partial \delta} &= \left(\frac{-2L}{\sin^2(\delta)} \times \sin\left(\frac{\gamma}{2}\right) \times \cos\left(\alpha_{z_k} + \frac{\gamma}{2}\right) + \frac{ds \cos(\delta)}{\tan(\delta)} \times \cos(\alpha_{z_k} + \gamma) \right) \times \cos(\alpha_{y_k})
\end{aligned} \tag{2.7}$$

$$\begin{aligned}
\gamma &= \frac{ds \sin(\delta)}{L} \quad \text{et} \quad c = \frac{2L}{\tan(\delta)} \sin\left(\frac{\gamma}{2}\right) \\
\frac{\partial Y_{v_{k+1}}}{\partial \alpha_{y_k}} &= -c \times \sin\left(\alpha_{z_k} + \frac{\gamma}{2}\right) \times \sin(\alpha_{y_k}) \\
\frac{\partial Y_{v_{k+1}}}{\partial \alpha_{z_k}} &= c \times \cos\left(\alpha_{z_k} + \frac{\gamma}{2}\right) \times \cos(\alpha_{y_k}) \\
\frac{\partial Y_{v_{k+1}}}{\partial ds} &= \cos(\delta) \times \sin(\alpha_{z_k} + \gamma) \times \cos(\alpha_{y_k}) \\
\frac{\partial Y_{v_{k+1}}}{\partial \delta} &= \left(\frac{-2L}{\sin^2(\delta)} \times \sin\left(\frac{\gamma}{2}\right) \times \sin\left(\alpha_{z_k} + \frac{\gamma}{2}\right) + \frac{ds \cos(\delta)}{\tan(\delta)} \times \sin(\alpha_{z_k} + \gamma) \right) \times \cos(\alpha_{y_k})
\end{aligned} \tag{2.8}$$

$$\begin{aligned}
\gamma &= \frac{ds \sin(\delta)}{L} \quad \text{et} \quad c = \frac{2L}{\tan(\delta)} \sin\left(\frac{\gamma}{2}\right) \\
\frac{\partial Z_{v_{k+1}}}{\partial \alpha_{y_k}} &= -c \times \cos(\alpha_{y_k}) \\
\frac{\partial Z_{v_{k+1}}}{\partial ds} &= -\cos(\delta) \times \cos\left(\frac{\gamma}{2}\right) \times \sin(\alpha_{y_k}) \\
\frac{\partial Z_{v_{k+1}}}{\partial \delta} &= \left(\frac{2L}{\sin^2(\delta)} \times \sin\left(\frac{\gamma}{2}\right) - \frac{ds \cos(\delta)}{\tan(\delta)} \times \cos\left(\frac{\gamma}{2}\right) \right) \times \sin(\alpha_{y_k})
\end{aligned} \tag{2.9}$$

$$\begin{aligned}
\gamma &= \frac{ds \sin(\delta)}{L} \quad \text{et} \quad c = \frac{2L}{\tan(\delta)} \sin\left(\frac{\gamma}{2}\right) \\
\frac{\partial \alpha_{z_{k+1}}}{\partial ds} &= \frac{\sin(\delta)}{L} \\
\frac{\partial \alpha_{z_{k+1}}}{\partial \delta} &= \frac{ds \cos(\delta)}{L}
\end{aligned} \tag{2.10}$$

Ici aussi se pose encore une fois le problème de la singularité en zéro pour les éléments $\frac{\partial X_{v_{k+1}}}{\partial \delta}$, $\frac{\partial Y_{v_{k+1}}}{\partial \delta}$ et $\frac{\partial Z_{v_{k+1}}}{\partial \delta}$. Comme précédemment, on résout par un développement limité au premier ordre :

$$\begin{aligned}
\frac{\partial X_{v_{k+1}}}{\partial \delta} &= \frac{-ds^2}{2L} \sin(\alpha_{z_k}) \cos(\alpha_{y_k}) \\
\frac{\partial Y_{v_{k+1}}}{\partial \delta} &= \frac{ds^2}{2L} \cos(\alpha_{z_k}) \cos(\alpha_{y_k}) \\
\frac{\partial Z_{v_{k+1}}}{\partial \delta} &= 0
\end{aligned} \tag{2.11}$$

$$(2.12)$$

Enfin, et pour clore cette étape, il faut définir le bruit de processus inhérent au phénomène modélisé. En effet, les informations proprioceptives ne permettent de venir mettre à jour que 4 des 6 composantes de position et d'orientation du véhicule. Par conséquent, et pour la suite, il convient de mettre un bruit sur les paramètres afin de leur donner la possibilité d'évoluer au cours du temps en marge du modèle. Sans cela, le filtre considèrera que leur valeur n'est assujettie à aucune incertitude : le filtre ne remet pas en cause ce dont il est sûr !

La définition de ce bruit de processus va se faire ici en tenant compte des composantes et capacités physiques de franchissement du véhicule. Il n'est pas question ici de traiter le problème du franchissement d'obstacle qui est un problème d'actualité (voir [Delmas 2011]), ni le problème de la définition d'un obstacle comme sur la figure 2.4.

Cependant, nous savons d'après les caractéristiques techniques du véhicule qu'il ne peut franchir une pente dont l'inclinaison est supérieure à d_{max}° . Par conséquent, nous allons utiliser cette information pour construire le domaine d'incertitude en position. Donc, si le véhicule se déplace et parcourt une distance ds , son altitude



FIGURE 2.4 : La notion d'obstacle est relative au véhicule.

possible est une combinaison de ds et de d_{max}° . Les équations de calcul du bruit de processus sont les suivantes et la représentation géométrique en est donnée dans la figure 2.5 :

$$\begin{aligned}
 \sigma_x &= ds \times \cos(\alpha_{z_k}) (1 - \cos(d_{max}^\circ)) \\
 \sigma_y &= ds \times \sin(\alpha_{z_k}) (1 - \cos(d_{max}^\circ)) \\
 \sigma_z &= ds \times \sin(d_{max}^\circ) \\
 \sigma_{\alpha_x} &= \text{faible devant } \sigma_{\alpha_y} \\
 \sigma_{\alpha_y} &= d_{max}^\circ \\
 \sigma_{\alpha_z} &= \text{faible devant } \sigma_{\alpha_y}
 \end{aligned} \tag{2.13}$$

2.3.2 Prédiction de l'observation

2.3.2.1 Introduction

Le modèle d'évolution permet de prédire une position mais ne permet pas de trouver les variations d'altitude comme nous l'avons vu précédemment. C'est l'observation qui vient combler cette déficience. Le processus d'observation nécessite une étape primordiale qui sera développée avant tout travail de projection quel qu'il soit : le choix et l'ordre de prise en compte des informations. Nous avons fait le choix, principalement pour des contraintes temps réel mais aussi dans un objectif de recherche par focalisation (cf. section 1.4.2), de considérer les informations de manière

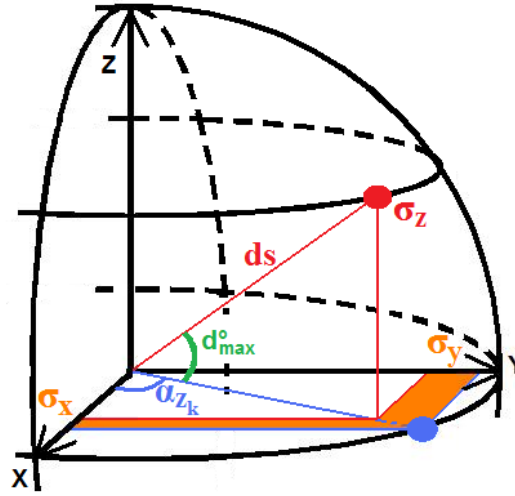


FIGURE 2.5 : La ligne bleue est l'estimation du chemin suivi entre deux instants tandis que la ligne rouge représente le chemin réel à la variation d'altitude maximale. L'écart-type de chaque axe est donné par la zone orange.

itérative. Les questions qui se posent à ce stade sont donc quel point prendre en premier et enfin, y a-t-il un ordre selon lequel les points doivent être choisis ? Pour trouver la réponse la plus logique, il faut se placer du point de vue de l'observateur : sachant que je souhaite réduire mon incertitude selon une dimension, quel est le couple point-observation qui résout mon problème ? De ce point de vue, on se rend compte que le choix ne peut être fait qu'à partir du moment où l'on dispose de tous les couples point-observation. C'est justement cela que l'on cherche à éviter avec le choix du filtre de Kalman itératif. Par conséquent, nous avons choisi d'aborder le problème en terme de divergence. Ainsi, on cherchera quelle dimension doit être mise à jour pour avoir le moins de couples point-observation erronés. Le problème de la sélection d'un point ne se pose pas en terme d'amélioration, mais plus en des termes de minimisation d'erreurs possibles suite à une mise à jour de la position ou de l'orientation. Une fois le critère de sélection établi et un point sélectionné, nous verrons comment définir dans l'image une zone de recherche à laquelle doit appartenir l'observation. Nous verrons comment la fonction d'observation impose une zone dans l'image trop restrictive et nous apporterons ici une contribution novatrice. Enfin, nous décrirons l'algorithme de recherche qui permet de trouver une observation correspondant à une prédiction.

2.3.2.2 Sélection d'un point à rechercher

En s'appuyant uniquement sur le filtre de Kalman, quelle que soit la version utilisée, on s'aperçoit que le point qui apporte la correction la plus importante est celui dont l'incertitude est la plus grande. Cependant, aucune information n'est donnée sur la ou les dimensions touchées par cette correction. L'approche de sélection choisie est décrite et argumentée dans les paragraphes suivants.

Pour commencer deux genres de paramètres sont en concurrence dans la localisation du véhicule : des angles (qui définissent l'orientation du repère) et des distances (qui définissent la position du repère). Ces deux entités, fondamentalement différentes, n'ont pas le même domaine de définition (\mathfrak{R}^+ pour les distances et $[-\pi; \pi]$ pour les angles). Comparer leurs incertitudes pour définir quel paramètre à besoin d'être mis à jour et définir un ordre de priorité devient complexe.

Il n'existe pas, à notre connaissance, d'articles ou publications faisant état de la comparaison de l'impact d'une quantité élémentaire appliquée sur les translations ou sur les rotations. Le principal problème se trouve au niveau de l'essence même de l'information. La translation est une donnée définie sur un espace ouvert de $-\infty$ à $+\infty$. Les angles sont définis sur un compact qui, pour des raisons de symétrie, sera défini comme allant de $-\pi$ à $+\pi$. Supposons maintenant une quantité élémentaire comme étant définie par exemple à 2. Quel que soit le système d'unité utilisé, il faudrait pouvoir dire qu'une translation de cette quantité élémentaire est définie comme une fonction de cette même quantité élémentaire associée à un angle. En des termes plus simples, il faudrait pouvoir comparer, dans notre cas, ce qui a le plus d'effet sur notre système entre une translation de 2 unités et une rotation de 2 unités.

Ce problème, en plus d'être complexe, dépend du système d'unité utilisé : les translations possèdent une multitude d'échelles qui peuvent inverser le phénomène. En exemple, une quantité élémentaire sur des kilomètres vis à vis d'angle en radian n'a pas le même poids qu'une quantité élémentaire sur des nanomètres, toujours vis à vis d'angle en radian. Dans notre cas, les angles sont en radians et les distances en mètres. Il apparaît cependant une évidence : une légère modification de la position du véhicule n'aurait que peu d'incidence dans l'observation des points lointains. Par contre, une légère modification de l'orientation aurait un effet notable sur les points, peut importe leur position par rapport à l'observateur. De plus, il peut être montré que la mise à jour avec des points éloignés affecte principalement l'orientation tandis qu'une mise à jour avec des points proches résout la localisation en position.

Compte tenu de ces dernières observations, nous avons choisi de sélectionner les points du plus éloigné au plus proche de l'observateur. L'utilisation des points éloignés nous permettra de réduire les incertitudes sur l'orientation dans un premier temps, les points proches terminant la mise à jour en positionnant le véhicule compte tenu de l'orientation préalablement sélectionnée. Sans pour autant avoir fait cette analyse, différentes stratégies ont été envisagées (par ordre d'ancienneté d'observation, du plus loin au plus proche, en alternance loin/proche, ...) et la méthode du plus loin au plus proche a été sélectionnée, en accord notamment avec les travaux de [Paz 2008].

Reste cependant un problème majeur : lorsqu'une observation des points éloignés est erronée. Notamment, si c'est le cas d'une des premières observations, tout le processus tombe en échec puisque l'orientation n'étant pas convenablement estimée, la recherche des autres informations ne sera pas cohérente. Pour résoudre ce cas, [Civera 2009] propose de réaliser un algorithme de RANSAC sur le premier point afin de ne pas choisir un mauvais couple candidat (*outlier*). Cependant, cette méthode

oblige encore une fois à calculer tous les couples point-observation (voir figure 2.6). Cela ne pose pas un réel problème avec un nombre de points limités à condition

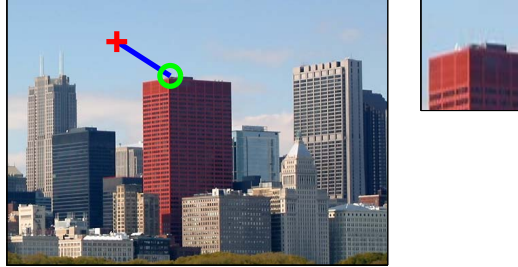


FIGURE 2.6 : La projection d'un point 3D dans une image donne le projeté (croix rouge). La recherche d'un descripteur correspondant (image en haut à droite) donne l'observation (cercle vert). La correspondance entre un point (ou son projeté) et une observation, symbolisé ici en bleu, forme un couple point-observation. Dans notre cas, le projeté d'un point correspond au projeté de sa meilleure estimation d'où l'équivalence de ce couple avec le couple estimation-observation que nous verrons plus tard.

aussi que la fréquence de la caméra soit suffisamment élevée pour que la projection des zones d'incertitudes ne couvre pas une zone trop importante de l'image. Aussi, la fréquence de la caméra contraint de ce fait la possibilité de répétition du motif ainsi que le temps de recherche de la meilleure observation pour un point donné. Etant donné que l'on souhaite s'affranchir de toutes ces recherches, nous avons choisi d'étudier les résultats de mise à jour des trois premiers points de manière indépendante. On cherchera alors à trouver, parmi ces points, ceux qui mènent à une orientation semblable (les deux plus proches). On reprendra alors le processus de recherche avec l'un des deux (ou des trois), dont la mise à jour est jugée correcte.

En reprenant l'exemple de la parabole évoqué dans 1.3.1.4, on peut dresser la comparaison entre trois approches : la méthode globale (présentée précédemment), la méthode itérative sans amélioration (aussi traitée dans le chapitre précédent) et la méthode itérative avec amélioration. Sur 100 échantillons, la méthode globale peut éliminer les données qu'elles jugent aberrantes vis-à-vis d'un modèle (ici une équation de parabole). La méthode itérative sans critère de choix peut éliminer des informations qu'elles jugent incompatible avec le modèle mais l'incertitude initiale étant importante, elle ne peut rejeter une aberration intervenant dans les premières données. Notre critère de sélection permet de traiter ce cas et d'éliminer une données aberrantes parmi les trois premières. Ce critère s'apparente à une méthode globale sur les trois premières observations. Les résultats obtenus suite à 1000 expériences de 100 échantillons sont consignés dans le tableau 2.1.

On constate une amélioration des résultats de la méthode itérative avec critère de choix par rapport à son homologue sans critère de choix. On remarque aussi que, dans tous les cas, la version globale reste la plus performante si ce n'est pour l'écart minimal (parmi toutes les réalisations, la méthode itérative avec critère de choix

Méthode	écart moyen	écart max	écart min
Itérative sans critère de choix	2.9173	9.2922	0.10818
Itérative avec critère de choix	1.2660	7.4925	0.00071
Globale	0.16642	0.6194	0.14296

TABLE 2.1 : Résultat d'estimation des paramètres de la parabole sur 1000 expériences de 100 valeurs en fonction de la présence ou non d'un critère de choix.

aboutit à une solution bien meilleure).

L'amélioration est d'autant plus significative que l'incertitude de départ est grande (l'état initial est $(0, 0, 0)$ et les variances associées sont $(25, 25, 25)$). De plus, la fonction d'observation étant non-linéaire, on retrouve un impact de la linéarisation sur les résultats. En effet, du fait de la sélection sur les trois premiers points, le point de linéarisation n'est pas le même. Pour aboutir à une meilleure estimation, il s'en suit un ensemble différent de données aberrantes détectées à une tolérance donnée. Remarquons enfin qu'une réalisation de l'expérience donne une solution bien meilleure que toutes celles fournies par la méthode globale (voir tableau 1.2 section 1.3.1.8).

À présent que le critère de sélection des points a été développé, attachons-nous à la prédiction de l'observation. Cette prédiction se définit par les relations qui donnent les coordonnées d'un point 2D dans l'image, projeté d'un point 3D de la carte de référence. La section suivante explique le modèle de caméra utilisé ainsi que le principe de projection et de calcul des coordonnées dans l'image.

2.3.2.3 Modèle d'observation

Le modèle d'observation est ici la relation qui lie un point 3D de l'espace avec sa projection dans l'image. Dans cet espace d'arrivée, les coordonnées ne sont plus exprimées sous forme métrique mais sous forme pixelique. Les pixels sont une discrétisation de l'espace image bornée par les limites du capteur (voir figure 2.7).

Le modèle pinhole (tête d'épingle) est le plus courant lorsqu'il s'agit de calculer la projection d'un point sur une image. Tous les rayons lumineux convergeant en un point (point focal, foyer ou centre optique), l'image se trouve donc, par définition, inversée et au-delà du point (Plan image). Cette inversion n'étant pas commode, les informations sont corrigées si bien que le plan image est déplacé en amont du foyer et l'image se retrouve à l'endroit comme expliqué sur la figure 2.8.

Pour réaliser une conversion correcte entre l'espace métrique et l'espace pixelique, les caméras doivent être calibrées de manière à déterminer les paramètres intrinsèques (propriétés propres au capteur) et extrinsèques (position du capteur sur le véhicule). Soit \mathbf{K} la matrice de paramètres intrinsèques définie comme suit :



FIGURE 2.7 : Illustration d'un capteur CCD Kodak : il contient 40 millions de pixels pour une taille de 40x33 mm (soit 6x6 μm par pixel).

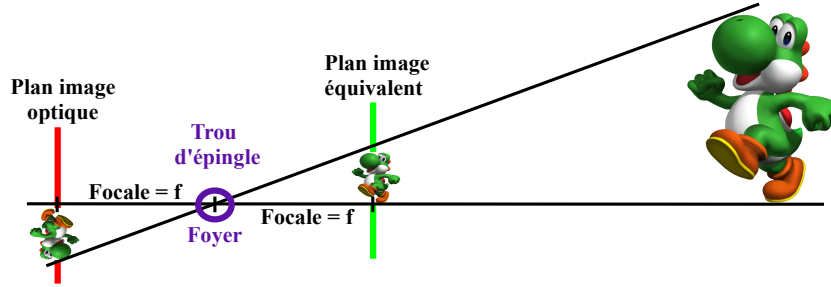


FIGURE 2.8 : Le modèle de projection pinhole (à travers un petit trou) place le plan image au delà du foyer. Il existe un plan équivalent en amont qui donne la même observation sans renverser l'image.

$$\mathbf{K} = \begin{pmatrix} 1 & 0 & 0 \\ C_u & 0 & f_u \\ C_v & f_v & 0 \end{pmatrix} \quad (2.14)$$

où (f_u, f_v) sont respectivement les focales suivant les axes u et v et (C_u, C_v) sont les coordonnées de la projection du point principal dans l'image dont les axes \mathbf{U} et \mathbf{V} sont définis comme sur la figure 2.9.

Soit $\mathbf{R}_{(\mathbf{v} \rightarrow \mathbf{c})}$ et $\vec{T}_{(v \rightarrow c)}$ la rotation et translation définissant le repère capteur \mathcal{R}_c dans le repère véhicule \mathcal{R}_v . Pour écrire les relations, nous aurons aussi besoin de la matrice de rotation $\mathbf{R}_{(m \rightarrow v)}$ et du vecteur de translation $\vec{T}_{(m \rightarrow v)}$, définissant le repère véhicule \mathcal{R}_v dans le repère monde \mathcal{R}_m :

$$\vec{T}_{(m \rightarrow v)} = (X_v, Y_v, Z_v)^T \quad (2.15)$$

$$\mathbf{R}_{(\mathbf{m} \rightarrow \mathbf{v})} = \mathbf{Rot}_z(\alpha_z) \mathbf{Rot}_y(\alpha_y) \mathbf{Rot}_x(\alpha_x) \quad (2.16)$$

où X_v, Y_v, Z_v sont les coordonnées du véhicule dans le repère monde et $(\alpha_x, \alpha_y, \alpha_z)$ sont respectivement les angles de roulis, tangage et lacet associés au véhicule, toujours par rapport au repère monde \mathcal{R}_m . Une illustration des repères en jeu est donnée

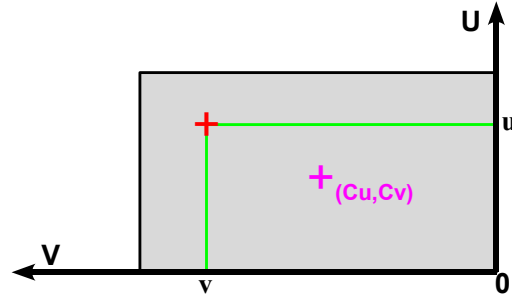


FIGURE 2.9 : Définition des axes du plan image.

par la figure 2.10. Le point 3D que nous nommerons ici $\vec{P}_t = (x_p, y_p, z_p)$ est défini

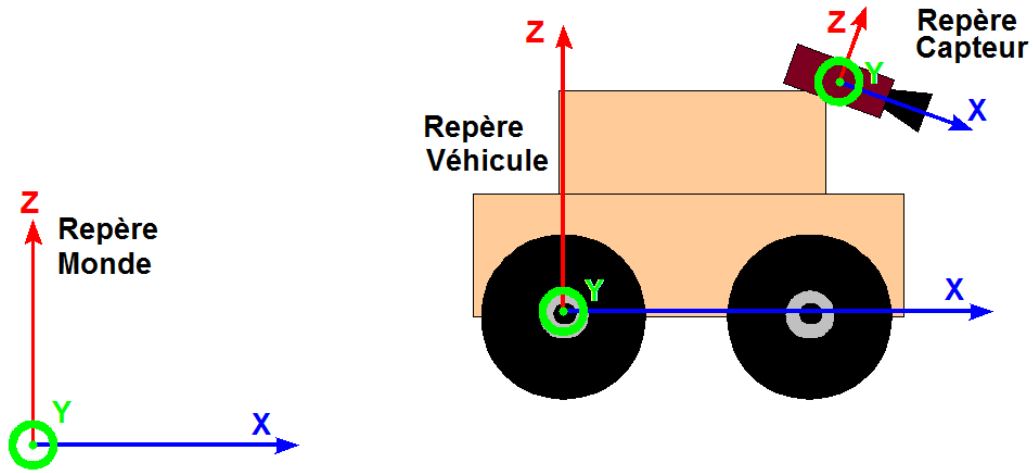


FIGURE 2.10 : Illustration des différents repères permettant de lier les informations définies dans le monde avec celles définies dans le repère du capteur.

dans le repère monde. Par conséquent, il doit être exprimé dans le repère capteur puis projeté dans l'image via la matrice de paramètres intrinsèques. On obtient la relation suivante :

$$\vec{P}_t^c = \mathbf{K} \mathbf{R}_{(v \rightarrow c)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}_t - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow c)}) = (x_c, y_c, z_c)^T \quad (2.17)$$

Cette relation fournit les coordonnées 3D du point dans l'espace capteur. Pour atteindre les coordonnées de l'espace image, il faut normaliser chaque composante relativement à celle sur l'axe de projection :

$$u = \frac{y_c}{x_c} \quad \text{et} \quad v = \frac{z_c}{x_c} \quad (2.18)$$

où x_c est dans ce cas l'axe de projection (voir figure 2.11)

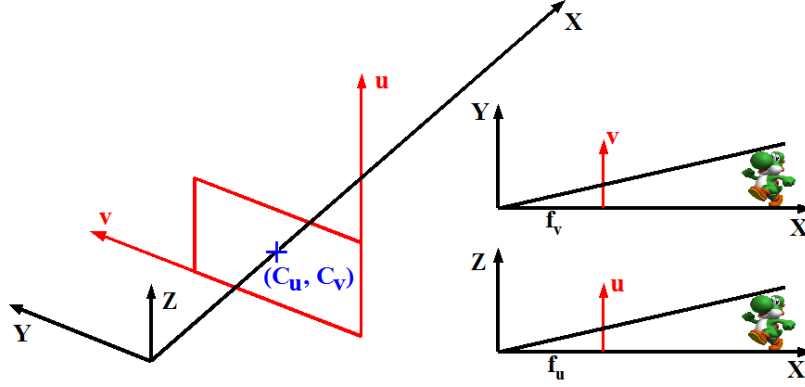


FIGURE 2.11 : A gauche, la représentation 3D des repères image (en rouge) et capteur (en noir). A droite, une projection sur les plans (Oxy) et (Oxz) permet de calculer indépendamment chacune des coordonnées images.

Nous avons déployé un formalisme matriciel car, comme nous le verrons, il permettra une formulation simple et élégante des calculs ultérieurs. Cette écriture, développée dans [Féraud 2010b], conduit aux relations suivantes :

$$u = \frac{\vec{K}_2 \mathbf{R}_{(v \rightarrow c)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow c)})}{\vec{K}_1 \mathbf{R}_{(v \rightarrow c)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow c)})} = \frac{Nu}{De} \quad (2.19)$$

$$v = \frac{\vec{K}_3 \mathbf{R}_{(v \rightarrow c)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow c)})}{\vec{K}_1 \mathbf{R}_{(v \rightarrow c)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow c)})} = \frac{Nv}{De} \quad (2.20)$$

où \vec{K}_l désigne la $l^{\text{ème}}$ ligne de \mathbf{K} ($l \in \{1, 2, 3\}$).

Nous avons donc les relations qui relient l'observation $\vec{y}_i = (\hat{u}_i, \hat{v}_i)^T$ du point $\vec{P}t$ avec le vecteur d'état $\vec{X}_{k|k}$ qui, rappelons-le, est uniquement constitué des 6 paramètres définissant la position et l'orientation du véhicule dans le repère monde.

Les équations 2.19 et 2.20 sont les équations du modèle d'observation que nous utiliserons par la suite. Cela implique :

$$\vec{y}_i = \mathbf{h}(\vec{X}) + \vec{\varepsilon}_i = \begin{pmatrix} u_i + \varepsilon_{ui} \\ v_i + \varepsilon_{vi} \end{pmatrix} \quad (2.21)$$

où (u_i, v_i) sont les mesures dans l'image correspondant à la prédiction $(\hat{u}_i, \hat{v}_i)^T$ compte tenu d'un bruit de mesure blanc et gaussien $\vec{\varepsilon}_i = (\varepsilon_{ui}, \varepsilon_{vi})^T$ (supposé de 1 pixel d'écart-type dans l'image dans notre cas).

2.3.2.4 Les jacobienes du modèle d'observation

L'utilité d'une écriture matricielle des équations 2.19 et 2.20 apparaît lors du calcul des jacobienes du modèle, nécessaires pour la propagation d'incertitude et

pour l'application du filtre de Kalman étendu. Observons dès à présent que l'observation d'un point n'est fonction que de la position du véhicule, de la caméra et du point en question $\vec{P}t_n = (x_n, y_n, z_n)$ (ici et dans ce qui suit le $n^{\text{ème}}$). On peut ainsi affirmer :

$$\frac{\partial \mathbf{h}(\vec{P}t_n)}{\partial \vec{P}t_{(j \neq n)}} = \mathbf{0} \quad (2.22)$$

La caméra étant fixe dans le repère véhicule, ses paramètres extrinsèques n'apparaissent pas dans le vecteur d'état. Ainsi, il faut calculer les jacobienues du modèle d'observation suivant les paramètres du véhicule et du point en question :

$$\mathbf{H}_{uv} = \begin{pmatrix} \frac{\partial u}{\partial X_v} & \frac{\partial u}{\partial Y_v} & \frac{\partial u}{\partial Z_v} & \frac{\partial u}{\partial \alpha_x} & \frac{\partial u}{\partial \alpha_y} & \frac{\partial u}{\partial \alpha_z} & \cdots & \frac{\partial u}{\partial x_n} & \frac{\partial u}{\partial y_n} & \frac{\partial u}{\partial z_n} & \cdots \\ \frac{\partial v}{\partial X_v} & \frac{\partial v}{\partial Y_v} & \frac{\partial v}{\partial Z_v} & \frac{\partial v}{\partial \alpha_x} & \frac{\partial v}{\partial \alpha_y} & \frac{\partial v}{\partial \alpha_z} & \cdots & \frac{\partial v}{\partial x_n} & \frac{\partial v}{\partial y_n} & \frac{\partial v}{\partial z_n} & \cdots \end{pmatrix} \quad (2.23)$$

Etudions le cas de $\frac{\partial u}{\partial X_v}$:

$$\frac{\partial u}{\partial X_v} = \frac{De \frac{\partial Nu}{\partial X_v} - Nu \frac{\partial De}{\partial X_v}}{De^2} = C_u \mathbf{R}_{(v \rightarrow c)}^T \mathbf{R}_{(m \rightarrow v)}^T \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \quad (2.24)$$

Avec $C_u = \frac{De \vec{K}_2 - Nu \vec{K}_1}{De^2}$. On en déduit alors $\frac{\partial u}{\partial Y_v}$ et $\frac{\partial u}{\partial Z_v}$:

$$\frac{\partial u}{\partial Y_v} = C_u \mathbf{R}_{(v \rightarrow c)}^T \mathbf{R}_{(m \rightarrow v)}^T \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \quad (2.25)$$

$$\frac{\partial u}{\partial Z_v} = C_u \mathbf{R}_{(v \rightarrow c)}^T \mathbf{R}_{(m \rightarrow v)}^T \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \quad (2.26)$$

En constatant que $\vec{T}_{(m \rightarrow v)}$ et $\vec{P}t_n$ joue le même rôle au sein du modèle d'observation, on en déduit que $\begin{pmatrix} \frac{\partial u}{\partial X_v} & \frac{\partial u}{\partial Y_v} & \frac{\partial u}{\partial Z_v} \end{pmatrix} = - \begin{pmatrix} \frac{\partial u}{\partial x_n} & \frac{\partial u}{\partial y_n} & \frac{\partial u}{\partial z_n} \end{pmatrix}$. Pour le calcul des jacobienues suivant l'orientation, il faut remarquer la chose suivante :

$$\frac{\partial \mathbf{R}_{(m \rightarrow v)}^T}{\partial \alpha_x} = \left(\frac{\partial \mathbf{R}_{(m \rightarrow v)}}{\partial \alpha_x} \right)^T = \left[\mathbf{R}_z(\alpha_z) \mathbf{R}_y(\alpha_y) \frac{\partial \mathbf{R}_x(\alpha_x)}{\partial \alpha_x} \right]^T \quad (2.27)$$

La relation 2.27 permet d'écrire par exemple $\frac{\partial u}{\partial \alpha_x}$:

$$\frac{\partial u}{\partial \alpha_x} = C_u \mathbf{R}_{(v \rightarrow c)}^T \left(\frac{\partial \mathbf{R}_{(m \rightarrow v)}}{\partial \alpha_x} \right)^T \left(\vec{P}t_n - \vec{T}_{(m \rightarrow v)} \right) \quad (2.28)$$

De la même manière, on en déduit $\frac{\partial u}{\partial \alpha_y}$ et $\frac{\partial u}{\partial \alpha_z}$. Les dérivées partielles de la coordonnée v sont calculées en remplaçant simplement C_u par $C_v = \frac{De \vec{K}_3 - Nv \vec{K}_1}{De^2}$.

L'intérêt de ce formalisme matriciel tient en trois points. Premièrement, il permet un passage simple des écritures d'une coordonnée à l'autre, en ne changeant qu'un seul coefficient. Il est donc tout à fait possible de calculer la partie commune une seule fois pour venir ensuite y appliquer le facteur propre à la dimension voulue. Deuxièmement, l'écriture des jacobiniennes est très simplifiée et le passage d'un gradient à un autre est facilité avec, encore une fois, une importante partie commune. Enfin, la représentation matricielle permet de s'affranchir de l'écriture des fonctions trigonométriques, épurant les relations au maximum. On peut, pour conclure cet aparté, donner l'équation de l'élément $(\frac{\partial u^2}{\partial x \partial y})$ de la hessienne sur la coordonnée u par exemple, avec ce formalisme :

$$\frac{\partial u^2}{\partial x \partial y} = \frac{\left[(K_2 - 2C_u D e) K_1 \mathbf{R}_{(m \rightarrow v)}^T \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} - K_1 \left(K_2 \mathbf{R}_{(m \rightarrow v)}^T \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \right) \right] \mathbf{R}_{(m \rightarrow v)}^T \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}}{D e^2} \quad (2.29)$$

2.3.2.5 Conclusion

A ce stade, nous disposons de tous les outils pour pouvoir projeter un point 3D depuis le repère monde jusqu'à dans l'espace image. Le calcul des jacobiniennes permettra aussi de pouvoir définir une zone de recherche par propagation de l'incertitude associée aussi bien au véhicule qu'au point en question.

Ce point 3D sera un point de la carte qu'il s'agira d'associer avec un point vu dans l'image et fera l'objet de la section suivante.

2.3.3 Recherche de l'observation - Association

2.3.3.1 Problème de l'association de points

Un point de la carte n'est pas uniquement défini par ses 3 coordonnées cartésiennes : elles ne suffiraient pas à le reconnaître dans l'image. Il est accompagné d'un descripteur et de la position du véhicule lors de la première observation. Le descripteur correspond au voisinage du point dans l'image comme le montre la figure 2.12.

C'est ce descripteur, aussi appelé patch ou imagerie, qui permettra de reconnaître le point dans l'image.

2.3.3.2 Zone de recherche d'un point dans l'image par l'approche classique

Nous disposons d'un point de la carte que nous allons rechercher dans l'image. Cette opération se traduit par la reconnaissance de son descripteur dans l'image. De plus, avant de se lancer dans une recherche du descripteur dans toute l'image, il est utile de restreindre la zone de recherche afin de limiter le temps de calcul de cette étape. L'idée principale consiste à calculer l'incertitude de la projection du point dans l'image (fonction de l'incertitude de la localisation du point et du véhicule) afin d'obtenir une zone de probabilité de présence qui sera, a fortiori, plus petite ou égale à l'image. Cette propagation d'incertitude se fait traditionnellement par le biais des jacobiniennes du modèle d'observation.

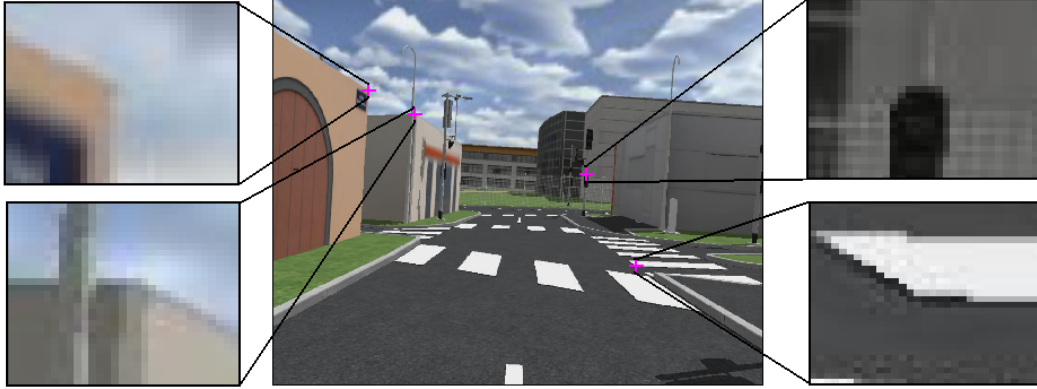


FIGURE 2.12 : Des points (croix magenta) sont extraites d'une image acquise par une caméra dans un environnement de simulation (source [Malartre 2011] et [Delmas 2011]). Pour chaque point, on peut extraire un descripteur (ici 15x15) qui correspond au voisinage du point dans l'image.

Soit \mathbf{P}_{Pt} la matrice de variance-covariance associée à un point 3D. On rappelle que le vecteur d'état est composé des degrés de liberté du véhicule $(X_v, Y_v, Z_v, \alpha_x, \alpha_y, \alpha_z)$ et que \mathbf{P} est la matrice d'incertitude associée. Par conséquent, on propage ces incertitudes à l'aide de la relation suivante :

$$\mathbf{P}_{2D} = \mathbf{H}_{\hat{X}} \mathbf{P} \mathbf{H}_{\hat{X}}^T + \mathbf{H}_{Pt} \mathbf{P}_{Pt} \mathbf{H}_{Pt}^T \quad (2.30)$$

où $\mathbf{H}_{\hat{X}}$ et \mathbf{H}_{Pt} sont respectivement les jacobiennes du modèle d'observation \mathbf{h} par rapport aux paramètres du vecteur d'état et par rapport au point courant. La figure 2.13 montre le résultat de la projection d'un ellipsoïde d'incertitude dans le plan image. D'une manière plus générale, [Ristic 2004] rappelle que dans le cas non-linéaire, le filtre de Kalman est sous-optimal. Si la carte est suffisamment précise, alors la différence entre un point de la carte et le vrai point du monde correspondant n'est pas toujours suffisante pour que la non-linéarité s'exprime. Toutefois, il convient d'aborder ce problème ici afin d'éliminer dès que possible les sources de divergences et de perte d'information.

La littérature explique que la linéarisation par le développement en série de Taylor au-delà du premier ordre est coûteuse en temps de calcul. C'est pourquoi elle est limitée au calcul des jacobiennes. Dans le cas de l'observation d'un point, la projection de l'ellipsoïde d'incertitude donne une ellipse dans l'image. Pour mieux percevoir l'action des jacobiennes de passage entre l'ellipsoïde 3D et sa projection 2D dans l'image, analysons le cas d'une incertitude 2D (équivalent à un ellipsoïde 3D dont la composante en altitude a été réduite à 0 et par conséquent, dans le plan du sol) et de sa projection, toujours par les jacobiennes, sur une droite (encore une fois, un plan dont la composante en altitude a été réduite à 0). La figure 2.14 illustre le propos.

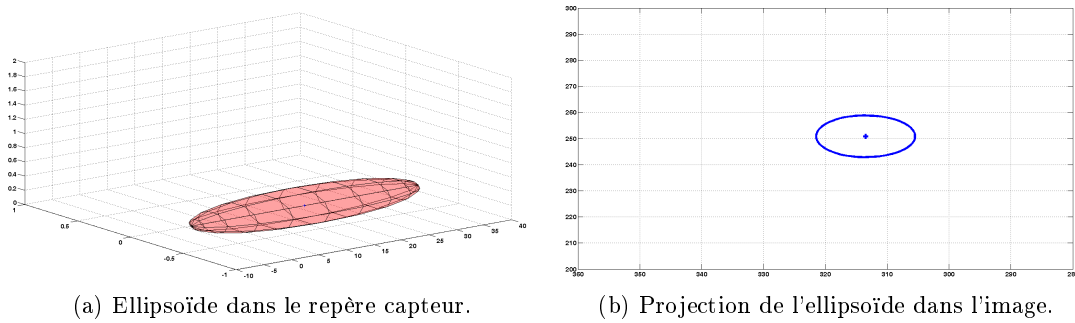


FIGURE 2.13 : Projection d'un ellipsoïde dans une image.

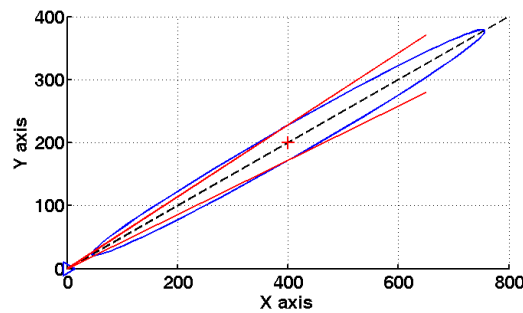


FIGURE 2.14 : Sur chaque graphique, l'observateur se situe à la position $(0, 0)$. L'incertitude (ellipse bleue) d'un point (croix rouge) est projetée sur une droite. L'incertitude obtenue permet de calculer un écart-type autour de la projection du point en question. De ces écart-types, on définit une zone qui, reprojetée dans le monde, donne l'ensemble des points dont la projection appartient à la zone d'incertitude délimitée par les deux droites rouges. On constate qu'une partie de l'incertitude n'appartient pas à ce cône.

On observe qu'une zone est oubliée par ce processus et cela reste valable avec les incertitudes 3D. Il serait plus convenable de calculer l'enveloppe de la projection, dans l'image, de tous les points de l'ellipsoïde 3D. La figure 2.15 montre la différence qu'il peut y avoir entre la projection des points situés sur l'ellipsoïde et la zone d'incertitude obtenue par le calcul des jacobiniennes.

On peut alors se rendre compte qu'une part non négligeable de l'espace d'incertitude est ignoré ce qui peut engendrer un échec de la phase de recherche.

2.3.3.3 Zone de recherche d'un point dans l'image : approche géométrique

Plutôt que de passer par une approximation par la linéarisation, pourquoi ne pas essayer de calculer l'enveloppe de la projection de l'ellipsoïde dans le plan image. En effet, plutôt que de calculer une linéarisation approximative, nous allons montrer que cette projection est une ellipse et que l'on peut calculer une zone rectangulaire

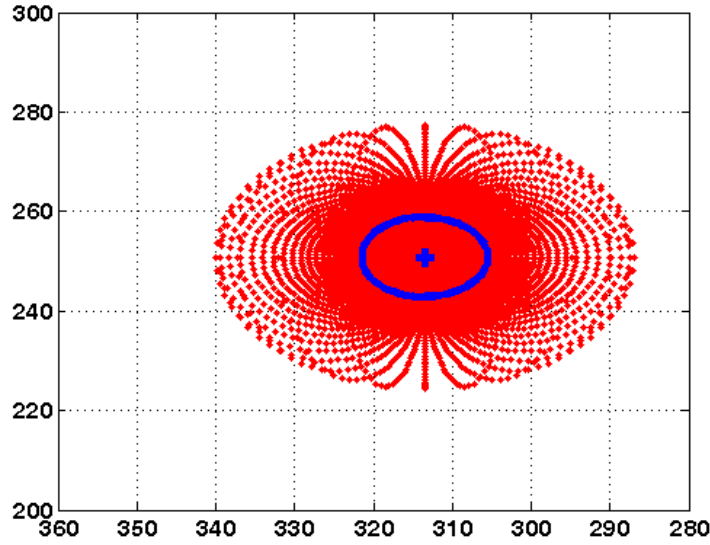


FIGURE 2.15 : La croix bleue représente la projection d'un point 3D dans l'image. L'ellipse bleue est la projection par les jacobienes de l'incertitude à 1σ associée à ce point. Chaque point rouge est la projection dans l'image d'un point situé sur l'ellipsoïde 3D à 1σ . On constate que la zone définie par les jacobienes est bien trop restrictive.

qui l'englobe de manière tangentielle. Etant donné que les propagations de repère en repère sont des fonctions linéaires, le problème vient de la projection de l'espace 3D au plan image. Par conséquent, nous allons traiter le problème directement dans le repère capteur où il ne reste plus que la fonction de projection à appliquer.

Énonçons le théorème 1 démontré par [Rebeix 1910] et le théorème 2 démontré dans l'Annexe A :

Théorème 1 *Un cône à base elliptique est coupé par tout plan sécant non issu du sommet ou non parallèle aux génératrices suivant une ellipse.*

Théorème 2 *Le lieu des points d'un ellipsoïde n'englobant pas l'origine, dont les plans tangents contiennent l'origine, est une ellipse.*

Grâce à ces deux théorèmes, on en déduit que l'enveloppe des projections des points de l'ellipsoïde dans l'image est une ellipse, ce qu'on peut voir en effet sur la figure 2.15. Cette ellipse est définie par 5 paramètres qui sont les deux coordonnées de son centre, les deux axes ainsi que l'orientation. Cependant, nous n'allons pas directement rechercher ces cinq paramètres mais nous allons nous appuyer sur une autre caractéristique de l'ellipse :

Pour tout point de l'ellipse, il existe un homologue tel que les tangentes en ces deux points soient égales.

Les formules qui vont suivre sont plus largement développées dans l'annexe B.

Soit un point $\vec{Pt_0} = (x_0, y_0, z_0)^T$ et la matrice d'incertitude associée $\mathbf{P}_{Pt_0} =$

$\begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix}$. Vérifions dans un premier temps que l'ellipsoïde 3D défini par \mathbf{P}_{Pt_0} n'englobe pas l'origine du repère. On calcule la distance de Mahalanobis suivante :

$$Dist_{\emptyset} = (\overrightarrow{Pt_0} - \overrightarrow{0})^T \mathbf{P}_{Pt_0}^{-1} (\overrightarrow{Pt_0} - \overrightarrow{0}) \quad (2.31)$$

Si $Dist_{\emptyset} \leq 1$ alors l'ellipsoïde englobe l'origine et il faut calculer le déplacement maximal d'un point dans l'image pour obtenir la zone de recherche cohérente. La section 2.3.3.4 étudie ce cas pour fournir les bornes maximales de la zone de recherche. Dans le cas où $Dist_{\emptyset} > 1$ alors l'ellipsoïde ne contient pas l'origine du repère et par conséquent, son projeté dans l'image peut être calculé et est contenu dans une ellipse.

Pour plus de commodité, notons $\mathbf{P}_{Pt_0}^{-1} = \begin{pmatrix} A & B & C \\ B & D & E \\ C & E & F \end{pmatrix}$. La méthode présentée ici propose de rechercher les points de l'ellipsoïde dont le plan tangent en ce point contient l'origine, et conduit à des tangentes horizontales ou verticales dans l'image. Pour ce faire, il faut d'abord définir la fonction qui place un point sur l'ellipsoïde. L'équation cartésienne d'un ellipsoïde est la suivante :

$$\mathcal{Q}(x, y, z) + \mathcal{L}(x, y, z) + \mathcal{C} = 0 \quad (2.32)$$

où \mathcal{Q} est une forme quadratique, \mathcal{L} est une forme linéaire et \mathcal{C} est une constante. Dans notre cas, la matrice de variance-covariance correspond à la forme quadratique et nous renseigne sur la taille de l'ellipsoïde ainsi que son orientation. Il serait possible, à partir des coordonnées du point $\overrightarrow{Pt_0}$ de remonter à la forme linéaire qui traduit un changement de repère, et la constante serait alors liée à l'écart-type de représentation de l'ellipsoïde. Cependant, on peut aussi remarquer que la distance de Mahalanobis donne directement le lien entre les points $\overrightarrow{Pt_T} = (x_T, y_T, z_T)$ appartenant à l'ellipsoïde et la matrice d'incertitude :

$$Dist_{Pt} = (\overrightarrow{Pt_T} - \overrightarrow{Pt_0})^T \mathbf{P}_{Pt_0}^{-1} (\overrightarrow{Pt_T} - \overrightarrow{Pt_0}) = 1 \quad (2.33)$$

$$\Rightarrow \mathfrak{E} : \begin{pmatrix} x_T \\ y_T \\ z_T \end{pmatrix} \mapsto \begin{pmatrix} x_T - x_0 \\ y_T - y_0 \\ z_T - z_0 \end{pmatrix}^T \begin{pmatrix} A & B & C \\ B & D & E \\ C & E & F \end{pmatrix} \begin{pmatrix} x_T - x_0 \\ y_T - y_0 \\ z_T - z_0 \end{pmatrix} - 1 \quad (2.34)$$

où \mathfrak{E} est la fonction permettant d'identifier un point de l'espace 3D comme appartenant ou non à un ellipsoïde d'incertitude donné.

De plus, on cherche des plans tangents en ces points particuliers. La normale d'une courbe ou d'une surface en un point est portée par le gradient de la fonction en ce point. Le gradient $\nabla_{\mathfrak{E}}$ de \mathfrak{E} en $\overrightarrow{Pt_T} = (x_T, y_T, z_T)$ est :

$$\overrightarrow{\nabla_{\mathfrak{E}}}(x_T, y_T, z_T) = \left(\frac{\partial \mathfrak{E}(x_T, y_T, z_T)}{\partial x_T} \quad \frac{\partial \mathfrak{E}(x_T, y_T, z_T)}{\partial y_T} \quad \frac{\partial \mathfrak{E}(x_T, y_T, z_T)}{\partial z_T} \right)^T \quad (2.35)$$

$$\overrightarrow{\nabla_{\mathfrak{E}}}(x_T, y_T, z_T) \equiv \begin{pmatrix} A & B & C \\ B & D & E \\ C & E & F \end{pmatrix} \begin{pmatrix} x_T - x_0 \\ y_T - y_0 \\ z_T - z_0 \end{pmatrix} \quad (2.36)$$

Par conséquent les plans qui contiennent l'origine et par extension le vecteur $\overrightarrow{Pt_T}$ sont définis par :

$$\left\langle \begin{pmatrix} x_T \\ y_T \\ z_T \end{pmatrix}, \overrightarrow{\nabla_{\mathfrak{E}}}(x_T, y_T, z_T) \right\rangle = \begin{pmatrix} x_T \\ y_T \\ z_T \end{pmatrix}^T \cdot \overrightarrow{\nabla_{\mathfrak{E}}}(x_T, y_T, z_T) = 0 \quad (2.37)$$

$$\Rightarrow \begin{pmatrix} x_T \\ y_T \\ z_T \end{pmatrix}^T \begin{pmatrix} A & B & C \\ B & D & E \\ C & E & F \end{pmatrix} \begin{pmatrix} x_T - x_0 \\ y_T - y_0 \\ z_T - z_0 \end{pmatrix} = 0 \quad (2.38)$$

Par ailleurs, on ne cherche pas n'importe quels plans tangents mais ceux qui conduisent à des tangentes horizontales ou verticales dans le repère caméra. En effet, puisque le plan image est défini de telle sorte que les axes U et V soient respectivement colinéaires aux axes Z et Y du repère caméra, alors les tangentes verticales de l'image sont des plans à tangentes verticales dans l'espace 3D et réciproquement. De plus, ces tangentes particulières qui conduisent à une boîte englobant l'ellipse dans l'image définissent une zone idéalement conçue pour les opérateurs de corrélation.

Etant donné que les plans tangents contiennent l'origine alors ceux conduisant à des tangentes horizontales contiennent aussi l'axe \overrightarrow{y} tandis que ceux conduisant à des tangentes verticales contiennent l'axe \overrightarrow{z} . Cela se traduit par une nouvelle équation pour chaque type de plan qui permet de définir les deux systèmes à résoudre :

$$\begin{aligned} \text{Points à tangentes verticales} & \left\{ \begin{aligned} & \begin{pmatrix} x_T - x_0 \\ y_T - y_0 \\ z_T - z_0 \end{pmatrix}^T \mathbf{P}_{Pt_0}^{-1} \begin{pmatrix} x_T - x_0 \\ y_T - y_0 \\ z_T - z_0 \end{pmatrix} = 1 \\ & (x, y, z) \cdot \overrightarrow{\nabla_{\mathfrak{E}}}(x_T, y_T, z_T) = 0 \\ & (x_T - x_0)C + (y_T - y_0)E + (z_T - z_0)F = 0 \end{aligned} \right. \quad (2.39) \\ \text{Points à tangentes horizontales} & \left\{ \begin{aligned} & \begin{pmatrix} x_T - x_0 \\ y_T - y_0 \\ z_T - z_0 \end{pmatrix}^T \mathbf{P}_{Pt_0}^{-1} \begin{pmatrix} x_T - x_0 \\ y_T - y_0 \\ z_T - z_0 \end{pmatrix} = 1 \\ & (x, y, z) \cdot \overrightarrow{\nabla_{\mathfrak{E}}}(x_T, y_T, z_T) = 0 \\ & (x_T - x_0)B + (y_T - y_0)D + (z_T - z_0)E = 0 \end{aligned} \right. \quad (2.40) \end{aligned}$$

$$\text{avec} \quad \overrightarrow{\nabla_{\mathfrak{E}}}(x_T, y_T, z_T) \equiv \begin{pmatrix} A & B & C \\ B & D & E \\ C & E & F \end{pmatrix} \begin{pmatrix} x_T - x_0 \\ y_T - y_0 \\ z_T - z_0 \end{pmatrix}$$

La résolution de ces deux systèmes donne respectivement les points conduisant à des tangentes verticales $\vec{P}t_{vi}$ et ceux conduisant à des tangentes horizontales $\vec{P}t_{hi}$ avec $i = \{1, 2\}$. Les détails étant précisés dans l'Annexe B, les résultats de chaque système sont les suivants :

$$\left\{ \begin{array}{l} \mathbb{M} = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \\ p_0 = -(x_0 \ y_0) \mathbb{M}_1 \quad \Bigg| \quad p_1 = -(x_0 \ y_0) \mathbb{M}_2 \\ r = (x_0 \ y_0) \mathbb{M} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} - 1 \\ \alpha = \left(1 \ -\frac{p_0}{p_1}\right) \mathbb{M} \begin{pmatrix} 1 \\ -\frac{p_0}{p_1} \end{pmatrix} \quad \Bigg| \quad \gamma = m_{22} \left(\frac{r}{p_1}\right)^2 - r \\ \beta = -2\frac{r}{p_1} \left(1 \ -\frac{p_0}{p_1}\right) \mathbb{M}_2 \\ \Delta = \beta^2 - 4\alpha\gamma \\ x_{vi} = \frac{-\beta + \sqrt{\Delta}}{2\alpha} \quad \Bigg| \quad x_{hi} = \frac{-\beta - \sqrt{\Delta}}{2\alpha} \\ y_{vi} = -\frac{p_0}{p_1} x_{vi} - \frac{r}{p_1} \quad \Bigg| \quad z_{hi} = -\frac{p_0}{p_1} x_{hi} - \frac{r}{p_1} \\ z_{vi} = z_0 - \frac{(x_{vi} - x_0)C + (y_{vi} - y_0)E}{F} \quad \Bigg| \quad y_{hi} = y_0 - \frac{(x_{hi} - x_0)B + (z_{hi} - z_0)E}{D} \end{array} \right. \quad (2.41)$$

Avec \mathbb{M}_i étant la $i^{ème}$ ligne de \mathbb{M} qui est respectivement définie dans le cas des points à tangentes verticales et horizontales par :

$$\mathbb{M} = \left(\begin{array}{cc} A - \frac{C^2}{F} & B - \frac{CE}{F} \\ B - \frac{CE}{F} & D - \frac{E^2}{F} \end{array} \right) \quad \Bigg| \quad \mathbb{M} = \left(\begin{array}{cc} A - \frac{B^2}{D} & C - \frac{BE}{D} \\ C - \frac{BE}{D} & F - \frac{E^2}{D} \end{array} \right)$$

Ces points, projetés dans l'image, donnent les limites de la zone de recherche rectangulaire qui majore la projection de l'incertitude. De plus, nous savons que ces points projetés sont sur une ellipse (voir théorèmes 1 et 2). L'intersection des segments formés par les couples de points aux tangentes identiques permet de trouver le centre de l'ellipse. En utilisant la position du centre ainsi que deux des quatre points (ceux dont les tangentes sont orthogonales), on peut reconstituer la forme quadratique associée à cette ellipse (la méthode est développée dans l'Annexe C).

Sur un déplacement simple avec un ellipsoïde conséquent, on obtient la figure 2.16 qui met en regard la projection par les jacobiennes et notre calcul de zone de recherche. On constate que le centre de l'ellipse générée par notre méthode n'est pas centrée sur la projection du centre de l'ellipsoïde. Cela est remarquable d'un point de vue statistique puisque cela signifie que la zone de recherche n'est pas centrée

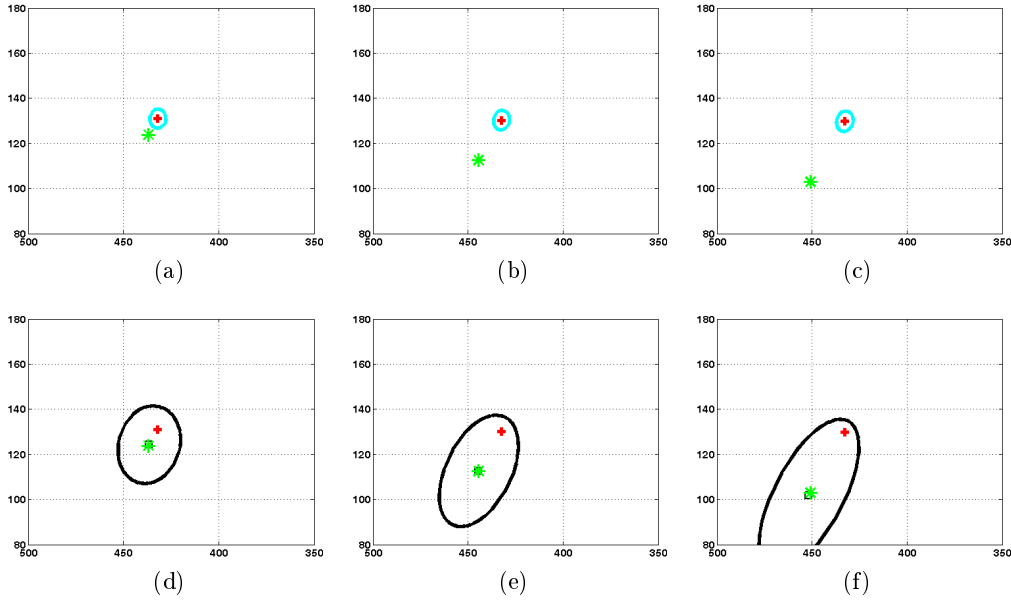


FIGURE 2.16 : Un point, réellement situé à 7 unités dans le repère monde, est estimé par un point de la carte qui lui se trouve à environ 100 unités avec une incertitude suffisamment grande pour garantir l'intégrité. Les projections de l'estimation dans l'image (croix rouge), de son espace d'incertitude (ellipses) et du point réel (étoile verte) sont représentées. En haut, la projection est réalisée par la méthode des jacobiniennes pour un déplacement de 0.3 unités (2.16a), de 0.7 unités (2.16b) et de 1 unité (2.16c). On constate alors la perte de l'intégrité puisque la projection du point réel n'appartient pas à la projection de l'incertitude de l'estimation. La méthode de projection développée pour les mêmes déplacements (respectivement 2.16d, 2.16e et 2.16f) reste intègre vis-à-vis de l'observation (l'étoile verte est contenue dans l'ellipse noire).

sur le maximum de vraisemblance. Cependant, tant que l'on s'assure que le point 3D réel est intègre vis-à-vis de l'estimation et de l'ellipsoïde d'incertitude associé, alors notre méthode garantit de trouver la projection du vrai point dans la zone de recherche dans l'image. Il est intéressant à ce stade de constater combien le centre de la zone peut être différent du maximum de vraisemblance. Par ailleurs, la complexité associée à cette étape supplémentaire est tout à fait acceptable. Ces travaux ainsi que le calcul de déplacement maximal qui suit ont fait l'objet de la publication [Bresson 2011].

2.3.3.4 Déplacement maximal d'un point dans l'image

Le calcul précédent permet, dans le cas où le repère de l'observateur n'est pas inclus dans l'ellipsoïde d'incertitude associé à un point, de trouver la projection exacte de l'espace d'incertitude dans l'image. Cependant, il n'est pas impossible que cet espace d'incertitude devienne incohérent et notamment dans le cas où l'on s'approche

de plus en plus de celui-ci. En effet, comme on peut le voir sur la figure 2.17a il arrive que cette zone limite soit aberrante du point de vue de la recherche d'observation. Il serait intéressant de connaître, compte tenu du déplacement et de l'erreur associée, le déplacement maximal que peut faire un point dans l'image. Théoriquement, puisque nous sommes dans un cas gaussien, ce déplacement est infini. Cependant, la Terre étant une sphère, on peut poser une limite : en prenant exemple sur la Tour Eiffel et ces 300 mètres de haut, on peut calculer qu'elle est théoriquement visible à partir d'une soixantaine de kilomètres (à 60 km elle mesure seulement 10 m). Par conséquent, en posant une limite de proximité de l'observation (de 1 mètre par exemple), on définit alors un segment limité sur le rayon de vue par essence infini.

On peut donc venir créer virtuellement 2 points sur un rayon de vue : un éloigné et l'autre proche. En tenant compte du déplacement subi et de l'erreur qui lui est assujettie, on peut calculer la projection dans l'image de ces deux points. L'écart que l'on mesure donne directement le déplacement maximal d'un point dans l'image. Enfin, si une zone de recherche est supérieure à cette limite, nous pouvons, sans trop d'erreur, la réduire. La figure 2.17 montre la réalisation du calcul ainsi que son impact sur une séquence d'images.

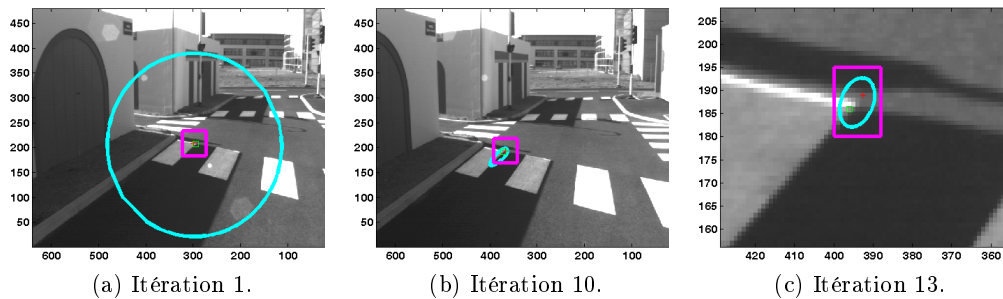


FIGURE 2.17 : La distance parcourue est calculée comme le quotient de la vitesse de déplacement (≈ 1 m/s) par la cadence de la caméra (10 Hz). Le principe consiste en la localisation d'un point au travers du filtre de Kalman étendu. La figure 2.17a illustre l'aberration de la zone de recherche (ellipse cyan) ainsi que la correction effectuée par le calcul de limite (cadre magenta). A l'itération 10 (après avoir parcouru 1 mètre), la figure 2.17b montre que l'incertitude est réduite et le calcul de limite reste cohérent dans cette phase. Enfin, 3 itérations plus tard, on constate, figure 2.17c, que la zone de recherche est bien conforme à la projection de l'incertitude et que le calcul de limite n'a pas été exécuté.

2.3.3.5 Recherche de l'observation : association de données

A présent qu'une zone de recherche a été délimitée, il faut reconnaître le point associé à une prédiction. Le descripteur du point prend alors toute son importance. En effet, reconnaître un point, c'est avant tout reconnaître une représentation de celui-ci qui le discrimine par rapport aux autres. Afin d'avoir des résultats optimaux,

il faudrait que l'extracteur de points, le descripteur et l'opérateur de recherche soient développés conjointement. Excepté sans doute pour le cas du SIFT (voir chapitre 3), cela n'est, à notre connaissance, pas le cas. En effet, la robustesse des extracteurs de points se définit aussi par leur faculté à reconnaître le même point comme un point d'intérêt dans différentes configurations. Mais en aucun cas, il ne permet de faire la correspondance.

Dans la littérature, apparaissent différentes méthodes de recherche des appariements que l'on peut scinder en deux classes : minimiser la différence ou maximiser la ressemblance. La minimisation des normes dites L_1 ou L_2 donne une première possibilité :

$$\text{Norme } L_1 = \sum_i \sum_j |I(i, j) - J(i, j)| \quad (2.42)$$

$$\text{Norme } L_2 = \sum_i \sum_j (I(i, j) - J(i, j))^2 \quad (2.43)$$

où $I(i, j)$ et $J(i, j)$ sont deux descripteurs à comparer. Comme on recherche une différence minimale, il faudra prendre comme résultat le minimum pour chaque norme. Une deuxième possibilité est donné par la maximisation de la ressemblance et la théorie des opérateurs de corrélation vient donner une solution non-optimale à ce problème désigné comme celui de l'association de données.

$$\text{Score de Cross-Corrélation : } CC = \sum_i \sum_j I(i, j) J(i, j) \quad (2.44)$$

$$\text{CC centré en zéro : } ZCC = \sum_i \sum_j (I(i, j) - \bar{I})(J(i, j) - \bar{J}) \quad (2.45)$$

$$\text{CC normalisé : } NCC = \frac{\sum_i \sum_j I(i, j) J(i, j)}{(N-1)\sigma_I \sigma_J} \quad (2.46)$$

$$\text{CC normalisé et centré : } ZNCC = \frac{\sum_i \sum_j (I(i, j) - \bar{I})(J(i, j) - \bar{J})}{(N-1)\sigma_I \sigma_J} \quad (2.47)$$

où N désigne le nombre d'éléments de $I(i, j)$ et de $J(i, j)$ et \bar{I} et \bar{J} sont respectivement les espérances calculées sur les ensembles $I(i, j)$ et $J(i, j)$. Etant donné que les descripteurs ont tous la même taille, ce coefficient $\frac{1}{N-1}$ sera retiré par la suite.

Parmi tous ces opérateurs, le ZNCC (Zero Norm Cross Correlation) est sans doute le plus utilisé en traitement d'image et en reconnaissance de descripteurs. Il est optimisé par rapport à une simple corrélation au travers de deux aspects : la normalisation qui règle les problèmes de dynamique comme ceux observés lors des variations de luminosité et le centrage autour de zéro qui uniformise la représentation. Pour comprendre l'opération réalisée par le ZNCC, il faut voir le descripteur du point d'une manière particulière. En effet, on peut considérer chaque pixel comme indépendant et leur donner à chacun un axe ou une dimension. Ainsi, un descripteur de 11x11 pixels peut se représenter sous la forme d'un vecteur dans un espace à 121 dimensions (soit 121 axes indépendants). Le produit scalaire de deux vecteurs \vec{v}_1 et

\vec{v}_2 est défini, en dimension finie, par :

$$\langle \vec{v}_1 ; \vec{v}_2 \rangle = \|\vec{v}_1\| \|\vec{v}_2\| \cos(\vec{v}_1 ; \vec{v}_2) \quad (2.48)$$

$$\Leftrightarrow \cos(\vec{v}_1 ; \vec{v}_2) = \frac{\langle \vec{v}_1 ; \vec{v}_2 \rangle}{\|\vec{v}_1\| \|\vec{v}_2\|} \quad (2.49)$$

Finalement, la valeur $\cos(\vec{v}_1 ; \vec{v}_2)$ traduit la ressemblance des deux vecteurs : plus ils se ressemblent, plus cette valeur tend vers 1. Dans le cas de nos descripteurs, nommons les I_1 et I_2 , l'opérateur ZNCC s'écrit de la manière suivante :

$$ZNCC(I_1, I_2) = \frac{\sum_i \sum_j [(I_1(i, j) - \bar{I}_1)(I_2(i, j) - \bar{I}_2)]}{\sqrt{\sum_i \sum_j (I_1(i, j) - \bar{I}_1)^2 \sum_i \sum_j (I_2(i, j) - \bar{I}_2)^2}} \quad (2.50)$$

En écrivant les descripteurs I_1 et I_2 comme des vecteurs colonnes dans l'espace à 121 dimensions, on peut alors ré-écrire la formule 2.50 :

$$\begin{aligned} ZNCC(\vec{I}_1, \vec{I}_2) &= \frac{\sum_k [(I_1(k) - \bar{I}_1)(I_2(k) - \bar{I}_2)]}{\sqrt{\sum_k (I_1(k) - \bar{I}_1)^2 \sum_k (I_2(k) - \bar{I}_2)^2}} \\ &= \frac{\langle \vec{I}_1 - \bar{I}_1 ; \vec{I}_2 - \bar{I}_2 \rangle}{\|\vec{I}_1 - \bar{I}_1\| \|\vec{I}_2 - \bar{I}_2\|} \end{aligned} \quad (2.51)$$

où $(\vec{I}_1 - \bar{I}_1)$ désigne, par abus de notation, la soustraction d'un scalaire à tous les éléments d'un vecteur. L'analogie entre les relations 2.49 et 2.51 permet de voir l'opérateur ZNCC comme un opérateur de ressemblance calculant le cosinus de l'angle orienté défini par les vecteurs $(\vec{I}_1 - \bar{I}_1)$ et $(\vec{I}_2 - \bar{I}_2)$. Toujours par analogie, on comprend que plus le résultat sera proche de 1, plus les vecteurs auront même direction et par conséquent, plus les descripteurs seront ressemblants. Un seuil reste tout de même à définir (il sera placé à la valeur de 0.8 dans notre cas), comme dans beaucoup d'applications de traitement d'image. Ainsi, une observation sera associée à un point s'il existe un point dans la zone de recherche de l'image tel que le résultat de l'opérateur ZNCC soit supérieur au seuil.

2.3.4 Mise à jour

L'étape de mise à jour consiste à utiliser le résultat de l'observation avec la prédiction pour venir préciser la pose de l'observateur. Dans le cadre du suivi et plus particulièrement du rejeu, on possède une carte d'amers 3D suffisamment précise. Il n'est, pour le moment, pas utile de venir l'enrichir ou de la préciser. Par ailleurs, le Visual SLAM étant intrinsèquement biaisé, ce problème est abordé dans une section ultérieure. Finalement, seuls les paramètres réglant la position et l'orientation de l'observateur sont consignés dans le vecteur d'état, à savoir les trois rotations et les trois translations.

Suite à chaque mise à jour successive un ultime test est effectué afin de s'assurer que le système ne diverge pas de manière générale. En admettant que l'incertitude propagée lors de l'étape de prédiction tienne compte de suffisamment de facteurs,

on peut s'attendre à trouver un résultat à l'intérieur de l'espace d'incertitude. Cela signifie qu'une mise à jour qui placerait le véhicule en dehors de l'espace d'incertitude de la prédiction ne doit pas être prise en compte car peut résulter d'une mauvaise association de données. Ainsi, la distance de Mahalanobis entre la position mise à jour et la position après la phase de prédiction, en regard de l'incertitude obtenue après l'étape de prédiction doit être inférieure à 1. Ce dernier seuil est une vérification a posteriori qui empêche le système de se mettre dans une configuration irrécupérable par la suite.

La totalité de la procédure est rappelée par l'algorithme 1 et par le schéma d'architecture figure 2.18.

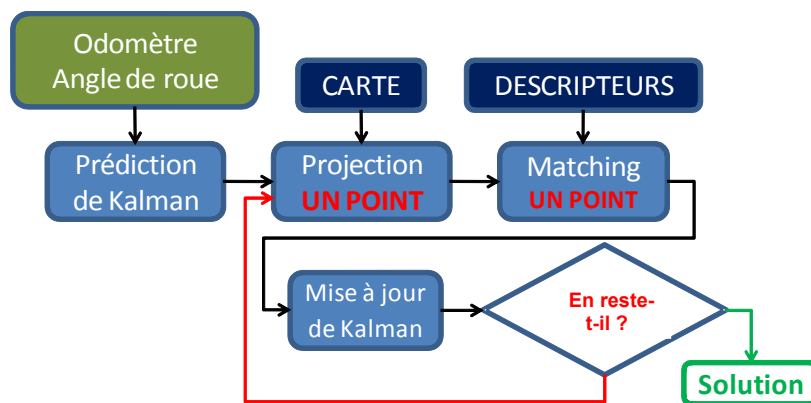


FIGURE 2.18 : Schéma du processus de la localisation en ligne

Algorithme 1 algorithme de localisation en ligne

Précondition : une image et les données proprioceptives relatives

- 1 : Prédiction de Kalman
 - 2 : Ordonnancement des points supposés observables
 - 3 : **Pour** chaque point **Faire**
 - 4 : Projection du point
 - 5 : Calcul de la visibilité
 - 6 : **Si** le point est visible **Alors**
 - 7 : Calcul des bornes
 - 8 : Matching
 - 9 : **Si** Le point est matché **Alors**
 - 10 : Mise à jour de Kalman
 - 11 : Vérification de l'intégrité
 - 12 : **Fin Si**
 - 13 : **Fin Si**
 - 14 : **Fin Pour**
 - 15 : Sauvegarde des données
 - 16 : **Retour** Localisation
-

2.3.5 Résultats

Dans un premier temps, les résultats présentés ici sont obtenus à partir d'une simulation. Le processus de création des données est simple et s'effectue par étape :

- on crée une trajectoire comme une succession de lignes droites et de courbes ;
- de cette trajectoire sont extraites les informations que fourniraient les capteurs proprioceptifs ;
- à partir d'un nombre maximum d'observations défini par image, on génère les points 3D de la carte ;
- pour chaque point, on associe la position de la première observation ainsi qu'un descripteur généré à partir d'un processus aléatoire.

Suite à cette étape, on obtient un jeu de données que l'on bruite afin de rendre le système plus proche de la réalité. Toujours en vue de travailler avec des données réelles, la contrainte de positionnement est abaissée à 5 cm (le couloir de navigation mesure donc 10 cm, centré sur la trajectoire). Les données de simulation sont montrées avec le tableau 2.2 et la figure 2.19.

Données	Vitesse	Cadence caméra	Cadence proprio
Valeurs	1 unité/seconde	10 Hz	20 Hz

Données	Distance totale	Dénivellation totale	Inclinaison maximale
Valeurs	1048 unités	+23.5 unités/−23.5 unités	15%

Données	Nombre de poses	Nombre de points	Descripteurs
Valeurs	31363	706	15 × 15 pixels

Erreurs	Odométrie	Angle de roue	Sur la carte
Valeurs	1%	1°	0.1 unités/dimension

TABLE 2.2 : Tableaux récapitulatifs des différentes variables de génération de l'environnement de simulation.

L'application du filtre de Kalman étendu avec prise en compte de l'optimisation de la zone de recherche sur une trajectoire simulée donne lieu aux tableaux 2.3. Les

Calcul des zones d'observation	écart moyen	écart maximal	écart minimal
Méthode projection	$2e^{-3}$ unités	$8e^{-3}$ unités	$4e^{-5}$ unités

Calcul des zones d'observation	Points vus	Points reconnus	Points utilisés
Méthode projection	≈ 16	≈ 15	≈ 14

TABLE 2.3 : Tableaux de comparaison des résultats.

résultats sur cette simulation, sans la prise en compte de l'optimisation de la zone de recherche, sont quasiment identiques à ceux avec l'optimisation et cela est dû

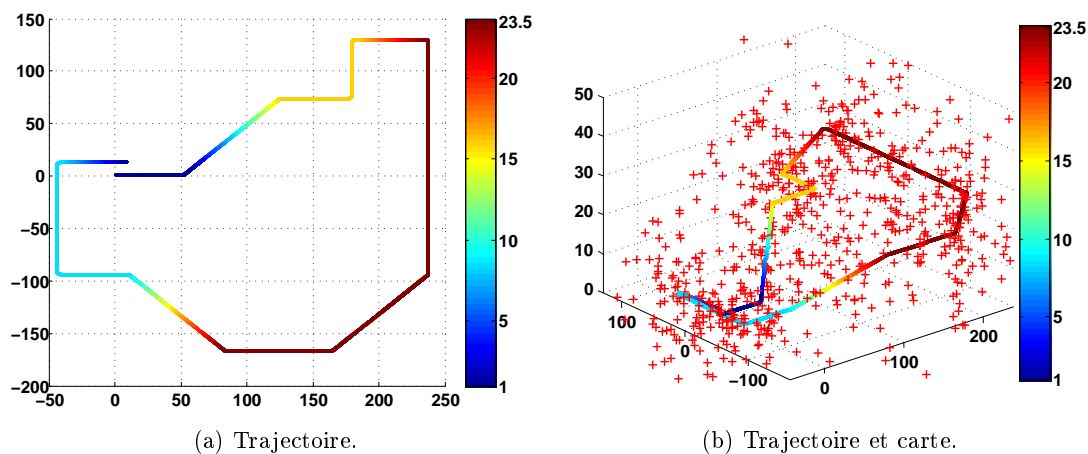


FIGURE 2.19 : Trajectoire de simulation : la figure 2.19a donne une vue de dessus de la trajectoire seule tandis que la figure 2.19b donne une vue 3D de la trajectoire avec les points de la carte générée. Dans chaque figure, la couleur des points de la trajectoire donne une information sur l'altitude (l'échelle des valeurs est à droite de chaque graphique).

principalement à l'incertitude des points. En effet, celle-ci étant faible (0.10 unités sur chaque dimension), les erreurs induites par la linéarisation ne sont pas clairement visibles. Comme on a pu le constater sur les figures 2.13 et 2.14, il faut que l'ellipsoïde d'incertitude soit grand : cela peut être le cas lors d'un SLAM tandis qu'un point est initialisé. A ce stade, il n'est concrètement pas possible de dégrader une simulation pour voir l'impact de la non-linéarité, sans dégrader du même coup la localisation. Partant du fait que la carte est connue, nous devons attendre le chapitre suivant.

Quoi qu'il en soit, la figure 2.20 montre l'écart en position de l'algorithme de localisation, avec optimisation des zones de recherche, par rapport à la position réelle simulée (distance euclidienne). On constate que l'intégrité est respectée (l'écart en position se situe dans la plage d'erreur) et que l'erreur de position reste inférieure à 0.05 unités. De plus, il s'agit d'une erreur de position estimée par rapport à la position réelle équivalente. Vérifier si le véhicule circule toujours dans le couloir de navigation demande un calcul d'écart latéral de la position estimée par rapport à la courbe de trajectoire. Cette erreur est encore plus faible que celle présentée dans la figure 2.20.

Des résultats sur des données réelles seront présentés dans le chapitre 5.

2.4 Localisation initiale

2.4.1 Problème de l'initialisation

Le cas de l'initialisation est un problème à part entière. Lors de l'initialisation, la position du véhicule est très imprécise vis à vis de la contrainte en position. De

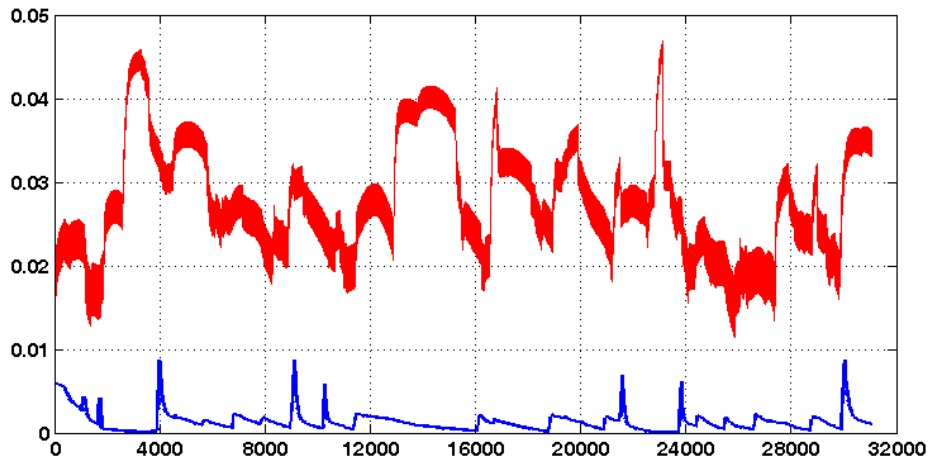


FIGURE 2.20 : Ce graphique montre l'écart en distance (en unités sur les ordonnées) en fonction de chaque itération (abscisse). En bleu, l'écart de chaque position fournie par l'algorithme de localisation par rapport à la position réelle de simulation. En rouge, l'incertitude à 1σ de cette distance. Le fait que la courbe d'écart ne dépasse jamais l'incertitude traduit l'intégrité du processus.

ce fait l'incertitude initiale est grande et, par conséquent, son projeté dans l'image devient suffisamment grand pour que l'association de données puisse être mise en échec. Un des cas de mauvaise association est celui de la redondance d'information dans l'image. Prenons le cas d'un passage piéton, les bandes blanches sont identiques et en nombre variable d'un passage à l'autre. La figure 2.21 montre le résultat de la recherche d'un coin de passage piéton sur l'ensemble d'une image. On observe, pour un seuil de ZNCC à 0.8, que le motif est redondant dans l'image.

2.4.2 Solution envisagée

La solution consiste en une exploration des différentes possibilités de mise à jour combinée au nombre d'associations détectées suite à cette mise à jour. Cette méthode rentre directement en conflit avec l'aspect temps réel mentionné comme contrainte dans le déroulement de l'application, c'est pourquoi elle n'est appliquée que pour résoudre l'ambiguïté de la phase d'initialisation. Un algorithme de RANSAC pourrait résoudre le problème de l'association mais nous avons préféré conserver une approche focalisante, notamment parce que la possibilité de mauvaises associations reste importante. En effet, un RANSAC impose d'avoir dès le départ un nombre suffisant de bonnes reconnaissances ce qui n'est pas forcément le cas ici. L'approche focalisante modifie les incertitudes si bien que les maxima de corrélation peuvent évoluer au fur et à mesure des mises à jour.

Malgré le fait que la position soit approximative, on peut trouver les points dont la première observation est suffisamment proche de la position de l'observateur

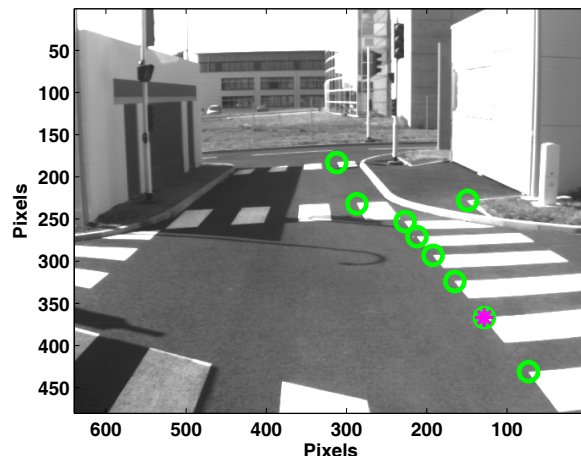


FIGURE 2.21 : Avec un seuil de recherche à 0.8 et un descripteur de taille 11×11 , les hypothèses d'observation d'un point (étoile magenta) sont données par les cercles verts. On constate ici la non-unicité du résultat.

pour qu'ils soient observables. Prenons le cas du premier point, une observation cohérente est recherchée dans une zone de l'image correspondant à la projection de l'incertitude du point en question et de celle de la pose du véhicule. Dans le cas où aucun correspondant n'est trouvé, le point est abandonné et on passe au suivant. Si une correspondance est trouvée, on calcule une mise à jour et le résultat est sauvegardé dans un duplicata du vecteur d'état. Il en va de même avec la covariance bien entendu. A ce stade, on possède une mise à jour potentielle qu'il est bien difficile de comparer avec d'autres. C'est pourquoi, une phase de recherche globale est amorcée. Pour chaque point autre que celui qui a servi à la mise à jour, on vient rechercher un correspondant dans une zone de recherche a fortiori plus petite car conséquente à une mise à jour. Le nombre de correspondances trouvées permet de caractériser le résultat. De plus, la distance entre chaque prédiction et chaque observation correspondante constitue une erreur de reprojection qui permet une discrimination plus fine. Ces paramètres sont enregistrés et on passe au point suivant en reprenant les informations d'état avant la mise à jour. Une fois tous les points passés en revue, on sélectionne celui qui possède le meilleur score puis on réitère à partir des informations d'état de la mise à jour qui lui correspond. De plus, on possède aussi toutes les observations ce qui évite l'étape de recherche globale. Chaque point ayant servi est retiré de la liste des points à étudier afin de ne pas introduire un phénomène de rumeur (ré-utilisation d'une même information pour conforter en apparence un résultat). Le processus est itéré tant qu'il y a des points dans la liste. Un schéma d'architecture de la procédure, par analogie avec la figure 2.18, est présenté dans la figure 2.22.

La procédure suivie est rappelée par l'algorithme 2. Encore une fois ici, les incertitudes sur les points et sur le véhicule ne sont pas suffisantes pour laisser s'exprimer la linéarité du modèle. De plus, rappelons qu'à ce stade, seuls les paramètres du vé-

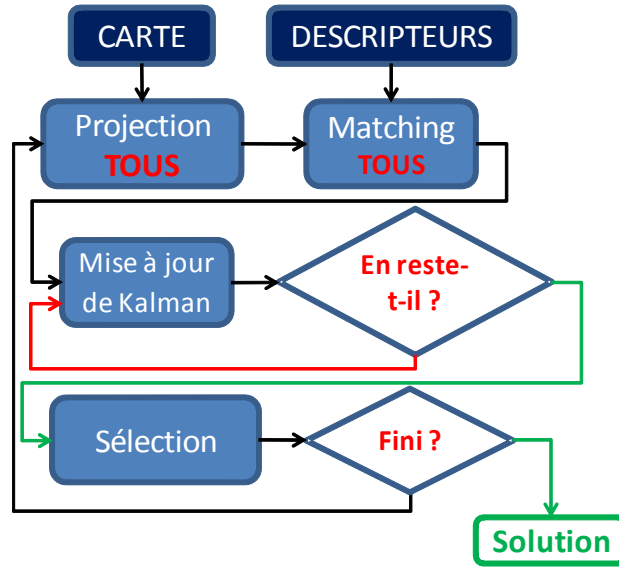


FIGURE 2.22 : Schéma du processus de la localisation initiale

Algorithme 2 Algorithme d'initialisation**Précondition :** Position origine supposée et incertitude intègre

- 1 : **Répéter**
- 2 : Reprojection de toutes les incertitudes dans l'espace image
- 3 : Recherche du meilleur critère d'association par ZNCC
- 4 : Création de couple 3D-2D
- 5 : Mise à jour de Kalman sur chaque couple indépendamment
- 6 : Nouvelle recherche d'association
- 7 : Décision du couple à garder
- 8 : Elimination du couple de la liste
- 9 : **Jusqu'à** Plus de couples **OU** on est au-dessus de la contrainte
- 10 : **Retour** (Localisation)

hicule appartiennent au vecteur d'état. Par conséquent, il faudrait une incertitude gigantesque qui traduirait une énorme méconnaissance de la pose initiale du robot. Dans le cas d'une utilisation du GPS en version civil, nous n'aurions qu'une incertitude de l'ordre d'une dizaine de mètres pour des angles relativement fiables.

2.4.3 Résultats de localisation initiale

Dans le cas de la localisation initiale, la représentation des résultats devient compliquée à interpréter. Par souci de simplicité, nous allons montrer les résultats de cette phase sur des données réelles. Disposant d'un banc stéréoscopique, on peut

réaliser une initialisation sur la caméra droite à partir de la position de la caméra gauche. Le montage des capteurs sur un CyCab¹ est illustré par la figure 2.23. La carte est ici constituée d'une cinquantaine de points dont seulement 19 sont observables depuis la position initiale et l'incertitude de chacun d'eux est de l'ordre du mètre sur chaque dimension. Etant donné qu'avec l'incertitude en position, la mise en correspondance peut changer, aucun point n'est rajouté au cours du traitement. La position des points n'est en aucun cas mise à jour et par conséquent, le vecteur d'état de cet algorithme ne contient, encore une fois, que les paramètres du véhicule. Le temps de calcul est important (3 secondes en moyenne sous MATLAB) et c'est pourquoi l'approche doit rester une approche d'initialisation (le temps de calcul étant essentiellement occupé par la phase de recherche d'association). Les résultats obtenus sont présentés dans la figure 2.24. En poussant le processus, on peut même



FIGURE 2.23 : Position des caméras (cercles rouges) et du GPS (cercle vert).

mener de front l'algorithme de localisation sur les deux caméras. La trajectoire de test est relativement courte (environ 5 m) et rectiligne mais elle suffit pour visualiser le comportement (voir figure 2.25).

Par ailleurs, le nombre de points matchés au terme de ce traitement est un bon indicateur de la pertinence du résultat ($>70\%$ par exemple). En effet, si, avec une incertitude faible sur la position du véhicule (puisque l'algorithme de localisation initiale réduit celle-ci), on est incapable de trouver des observations dans les zones d'incertitudes, alors l'algorithme a échoué. Cette approche permet d'avoir une position et une incertitude initiale adaptée à l'algorithme de localisation tout en testant de multiples hypothèses de mise à jour.

Nous avons apporté une solution de localisation temps réel qui s'appuie sur une connaissance a priori de la position de véhicule, sur les données proprioceptives ainsi que sur les images de l'environnement. Les images permettent de reconnaître des amers préalablement traités et enregistré dans une carte de référence. Dans le cas où la localisation a priori est imprécise (de l'ordre du mètre pour la position et de la dizaine de degrés pour l'orientation), nous avons donné une solution, plus couteuse en temps de calcul, pour résoudre le problème. Cette fonction de localisation sera testée sur des données réelles dans le chapitre 5. En attendant, nous allons aborder le second volet du projet à savoir la modification de la trajectoire par un opérateur. En

1. Cycab : petit véhicule électrique distribué par la société Robosoft.

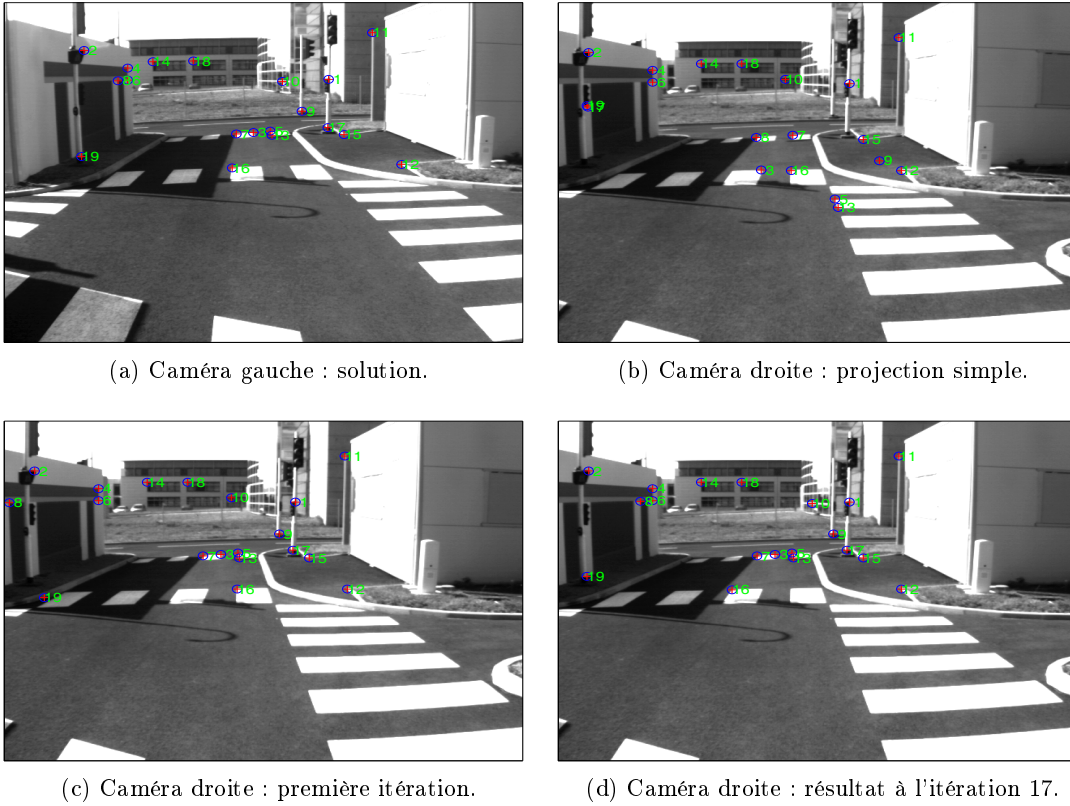


FIGURE 2.24 : Résultats de la phase d'initialisation d'une des caméras d'un banc stéréo. **2.24a** : image de la caméra gauche, la projection des points 3D (croix rouges cerclées en bleu) est labellisée (en vert). **2.24b** : image de la caméra droite, résultat de la première extraction d'information par corrélation. **2.24c** : image de la caméra droite, résultat au terme de la première itération, la solution d'un couple a été sélectionnée et appliquée. **2.24d** : image de la caméra droite, résultat de l'algorithme après 17 itérations (le résultat était trouvé, dans ce cas, depuis l'itération 3). On peut constater qu'un point est mal positionné (le numéro 10) : cela est dû à la redondance du motif. Il sera éliminé lors des mises à jour divergentes.

effet, suite à la présence d'obstacle par exemple, le véhicule peut être soudainement télé-opéré sur une portion de chemin qui n'a pas été cartographié. Nous verrons dans le chapitre suivant les problèmes que cela impliquent et nous donnerons une méthode pour traiter cette difficulté.

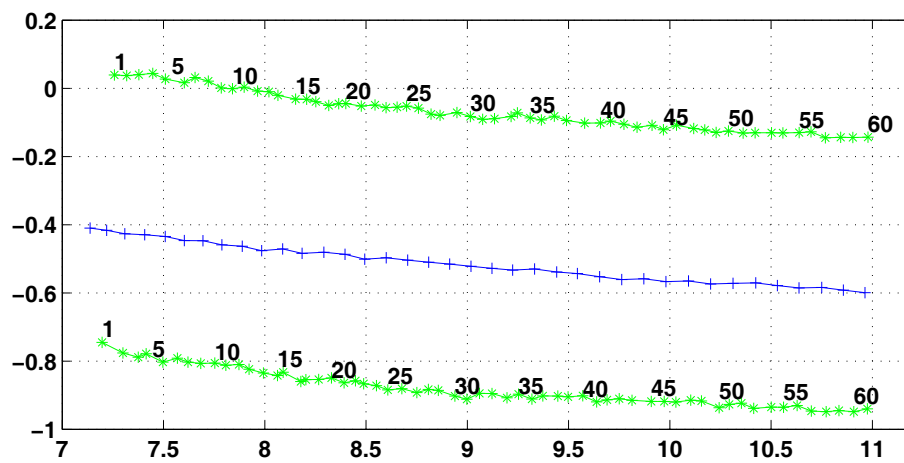


FIGURE 2.25 : illustration de la localisation simultanée des deux caméras d'un banc stéréo traitées indépendamment en vue de dessus. On constate que leur localisation relative est correcte sur une courte distance (courbe en vert et labellisée) et cohérente par rapport à la position du véhicule (courbe bleue).

SLAM par filtrage de Kalman étendu

Sommaire

3.1	Introduction	70
3.2	Initialisation de nouveaux points 3D	71
3.2.1	Extraction de points	71
3.2.2	Création du point 3D	76
3.2.3	Création de la matrice d'incertitude associée	77
3.3	Prédiction	86
3.4	Mise à jour	88
3.4.1	Mise à jour avec la carte de référence	88
3.4.2	Mise à jour avec la carte SLAM	89
3.5	Problème de linéarisation du modèle d'observation	89
3.5.1	Cas d'étude 2D : $\mathbf{z} = \mathbf{y}/\mathbf{x}$	89
3.5.2	Calcul de la correction	93
3.5.3	Résultat de la correction	95
3.6	Retour au Visual SLAM	96
3.7	Conservation d'un point	100
3.8	Résultats de simulation	102
3.9	Conclusion	104

3.1 Introduction

Le chapitre 2 a montré une réalisation d'une fonction de localisation fondée sur une carte 3D du monde déjà disponible, ainsi que sur des capteurs proprioceptifs et une vision monoculaire. Deux versions ont été développées permettant d'obtenir un positionnement d'une part, à partir d'une connaissance a priori (processus en ligne et temps réel), et d'autre part à partir d'une incertitude plus importante (processus d'initialisation long). L'algorithme dit « en ligne » fournit une localisation précise avec une carte peu dense ce qui permet de stocker de longs trajets. Dès lors, nous sommes capables de rejouer un chemin pré-enregistré et dont l'environnement a été cartographié.

Cependant, cet algorithme ne permet pas de s'écarter de la trajectoire, ce qui signifie qu'aucune prise en main d'opérateur n'est possible pour éviter ou redéfinir la trajectoire de référence suite à une modification de l'environnement ou tout autre intervention. Le fait de vouloir s'éloigner de la trajectoire de référence, pour probablement y revenir par la suite, suppose de pouvoir, à son tour, cartographier l'environnement. Cette problématique est, à présent, parfaitement en cohérence avec les approches SLAM. Il vient rapidement à l'esprit que le véhicule va donc devoir ajouter, de son propre chef, de nouveaux points pour enrichir sa carte dans les zones qui n'ont pas été définies dans la carte de référence. Par ailleurs, il n'est pas impensable d'envisager que cette carte de référence a elle-même été construite par un véhicule, enregistrant du même coup sa trajectoire. De ce point de vue, sa carte SLAM, ainsi que sa trajectoire deviendront respectivement la carte de référence et la trajectoire de référence pour ceux qui souhaiteraient rejouer le chemin. Ce point sera principalement abordé lors des essais réels développés dans le chapitre 5.

Rappelons qu'une vision monoculaire ne permet pas d'obtenir, avec une seule observation la position 3D d'un point. De plus, le véhicule n'étant pas forcément très rapide, utiliser deux images successives pour faire une triangulation pose un problème de parallaxe. Il serait possible de traquer le point dans l'image jusqu'à ce que la parallaxe soit suffisante pour opérer une triangulation mais il est alors dommage de perdre les informations de suivi du point. C'est pourquoi, il paraît plus pertinent de faire converger une estimation de manière itérative à partir de l'information fournie par chaque observation du point. Cela fait donc émerger un problème récurrent dans ce genre d'approches : comment initialiser un point et quelle incertitude lui associer ?

Par ailleurs, le SLAM par filtrage de Kalman est un processus biaisé [Chanier 2010] et ce biais¹ dépend non seulement de l'algorithme en lui-même mais aussi des capteurs et de leur position sur le véhicule. De cela on peut en déduire que le biais est unique pour chaque système. Notons par ailleurs que si le biais est unique, il n'est pas forcément reproductible et c'est ce qui rend les phases de calibration si délicates.

Malgré cela, nous nous apprêtons à faire un SLAM hybridant deux cartes : une

1. On utilisera le terme de biais ici et par la suite pour signifier la dérive classique du processus SLAM.

carte SLAM qui est propre au véhicule navigant à l'aide d'une carte de référence, elle-même construite par un autre véhicule. Par conséquent, la dérive de chacune des cartes n'est pas forcément la même. Cela implique un biais de l'une par rapport à l'autre. Il faut donc tenir compte de ce biais pour pouvoir trouver des correspondants des points de la carte de référence dans l'image après avoir quitté la trajectoire et navigué sur la base de notre propre carte. Ces différents aspects ainsi qu'une attention toute particulière à la non-linéarité du modèle d'observation seront abordés dans les sections suivantes.

3.2 Initialisation de nouveaux points 3D

On cherche ici à créer de nouveaux points 3D de manière à initier le processus SLAM. Créer une carte, c'est créer des points 3D en concordance avec les observations qui leur sont associées. Le choix de la vision monoculaire ne permet pas d'obtenir directement la profondeur liée à une observation. C'est par la méthode itérative du filtre de Kalman qu'elle sera estimée en exploitant les données proprioceptives. Mais tout d'abord, il faut choisir quels points de l'image sont susceptibles de devenir des amers. Cela fera l'objet de la partie 3.2.1. Par la suite, il faut initialiser un point 3D avec sa matrice d'incertitude en cohérence avec l'observation courante. Cette phase sera abordée en 3.2.2. Enfin, l'algorithme de localisation sera déroulé comme il l'a été dans le cas de la localisation avec la carte de référence. Nous verrons plus précisément les impacts et les différences de comportement entre le traitement des points de la carte de référence et les points créés.

3.2.1 Extraction de points

3.2.1.1 Introduction

Si on demande à deux individus de sélectionner un nombre défini d'informations dans une image, les résultats ne seront pas parfaitement identiques. Cependant, si on leur présente ultérieurement la même image, ils sauront la reconnaître. L'extraction d'information est propre à chaque individu, de même que le processus de reconnaissance. Dans le même esprit, extraire des points d'intérêt d'une image est un processus suffisamment complexe pour qu'il n'y ait pas une solution universelle : que retenir et sous quelle forme, telles sont les questions à se poser.

Tout d'abord, qu'est-ce qu'un point d'intérêt ? Il s'agit d'un point défini dans l'image qui peut être détecté de manière robuste. C'est-à-dire que, dans son voisinage proche, il est remarquable par rapport aux autres. Cette spécificité peut être de nature physique comme les sommets d'un triangle ou les coins d'un immeuble, de nature colorimétrique, comme un coquelicot sur un gazon, ou d'une intensité particulière comme les étoiles dans le ciel nocturne.

Détecteurs	Description	Type	Multi-échelle
Harris	La réponse est calculée à partir de la matrice des moments du second ordre de chaque pixel, elle-même calculée à partir des dérivées premières [Harris 1988].	Extracteur de coins	Non, mais une variante existe [Mikolajczyk 2004a]
Shi-Tomasi	L'approche s'appuie sur le même principe que le détecteur de Harris avec un critère de sélection plus restrictif [Shi 1994].	Extracteur de coins	Non, mais une variante existe [Mikolajczyk 2004a]
FAST	Features from Accelerated Segment Test : cette méthode étudie la configuration des niveaux de gris des pixels situés sur le périmètre d'un cercle centré sur le point considéré [Rosten 2006].	Extracteur de coins	Non
LOG	Laplacian Of Gaussian : le résultat repose sur le laplacien des niveaux de gris obtenu par une différence de gaussiennes [Mikolajczyk 2004b].	Extracteur de zones	Oui
DOG	Difference Of Gaussian : le principe est équivalent à celui d'un filtre passe-bande [Lowe 2004].	Extracteur de zones	Oui

TABLE 3.1 : Tableaux des principaux extracteurs de points d'intérêt.

3.2.1.2 Extracteurs de points

Il existe de nombreux extracteurs de points, chacun étant spécifique à un type de point, que l'on peut toutefois catégoriser : les détecteurs de coins et les détecteurs de zones. Les détecteurs de Harris [Harris 1988], Shi-Tomasi [Shi 1994], FAST [Rosten 2006], LOG [Mikolajczyk 2004b] et DOG [Lowe 2004] sont rassemblés en une liste non exhaustive dans le tableau 3.1.

La robustesse et la fiabilité des extracteurs de points sont définies sur leur capacité à extraire les mêmes points sur une image modifiée en rotation, changement d'échelle ou d'illumination. A ce niveau, les corrélateurs et autres algorithmes d'association n'entrent pas en ligne de compte.

Il existe une autre catégorie, un peu plus en marge des deux proposées car moins spécifique aux images, il s'agit des extracteurs de points sur la base d'informations statistiques comme l'analyse de texture de Haralick [Haralick 1979]. L'analyse de texture n'est pas l'apanage des approches statistiques puisque la description du voisinage sur laquelle sera fondée la recherche et la reconnaissance du point demeure cruciale. L'algorithme SIFT [Lowe 1999] (Scale Invariant Feature Transform), bien que coûteux en temps de calcul, présente l'avantage de contenir un extracteur de point associé à un descripteur. Les méthodes de recherche des descripteurs dans une image ne sont généralement pas optimales pour le descripteur choisi. On peut

montrer que l'opérateur de corrélation, largement utilisé comme score de similitude, est optimal pour un descripteur analogue à un bruit blanc. Or les extracteurs de points d'intérêt qui seraient caractéristiques de ce type de signal entreraient dans la classe des extracteurs statistiques qui sont connus pour ne pas être coûteux en temps de calcul. On ne se propose pas ici de résoudre le problème d'optimalité entre un descripteur et un corrélateur car, pour des zones de recherches suffisamment petites, l'absence de redondance d'information limite l'apparition de mauvaises associations.

Lors de travaux antérieurs menés chez Thales-Optronique, une approche a été étudiée sans pour autant être intégrée au système final. On peut remarquer que le détecteur de Harris travaille sur les gradients de l'image ce qui le rend invariant aux changements d'illumination. Cette dérivée introduit un bruit haute fréquence qui justifie l'utilisation d'un lissage gaussien. Cependant, il reste toutefois défaillant sur les approches multi-échelles. Par ailleurs, le DOG a la caractéristique d'être multi-échelle sans être robuste aux changements d'illumination. L'approche étudiée était de réaliser un extracteur DOG (Difference Of Gaussian, voir le tableau 3.1) non pas directement sur l'image, mais sur le gradient de l'image. Ainsi, la succession de lissage gaussien est plus efficace dans l'élimination du bruit haute fréquence tandis que les gradients rendent l'extracteur moins sensible aux variations d'illumination. Le temps de calcul étant supérieur à celui d'un DOG ou d'un Harris, malgré une robustesse plus intéressante, il n'a pas été retenu pour la suite du développement.

Dans la littérature, on trouve le détecteur de Harris comme étant le plus utilisé avec des codages différents notamment sur la fonction de seuillage suite au score calculé. Citons [Malartre 2011] qui a obtenu de bons résultats de SLAM sans pour autant choisir un extracteur de points en particulier. Aussi, notre choix s'est porté sur le plus commun : le détecteur de Harris. Cependant, compte tenu de l'application visée, une fonction de seuillage et une segmentation de l'image ont été réalisées afin d'obtenir une plus grande rapidité.

Le détecteur de Harris est facilement parallélisable puisque le calcul du score relatif à un pixel compte tenu de son voisinage n'est pas dépendant du calcul du score d'un autre pixel. Ainsi, diviser l'image en sous-images pour appliquer sur chacune l'extracteur de Harris semble à première vue une bonne idée pour gagner en temps de traitement (voir l'algorithme 3).

Algorithme 3 algorithme de Harris par découpage

Précondition : Une image et un vecteur de découpage

- 1 : Calcul du nombre de point (Npts) par sous-image
 - 2 : **Pour** Chaque sous-image **Faire**
 - 3 : Calcul des scores de HARRIS
 - 4 : Extraction des Npts scores maximaux
 - 5 : **Fin Pour**
 - 6 : **Retour** Les points de Harris extraits
-

Cependant, cela induit une non-homogénéité des résultats comme le montre les figures 3.1a et 3.1b. Cette différence provient du fait que les maxima ne sont plus

triés sur l'ensemble de l'image, mais uniquement sur la sous-image. Il serait possible de rassembler tous les scores pour les trier ensemble mais pourquoi ne pas éliminer des points qui, de toute façon, ne sont pas porteurs d'information.

C'est plutôt dans cette optique que nous allons travailler, d'autant plus que l'on connaît le nombre de points que l'on cherche. Soit N le nombre de points recherchés et l'image divisée en m lignes et n colonnes. Nous devons donc chercher $\text{ceil}\left(\frac{N}{m \times n}\right)$ points dans chaque sous-image avec la fonction $\text{ceil}(x)$ définissant l'arrondi à l'entier supérieur ou égal à x . De ce fait, au moins un point par sous-image sera recherché et cela nous donne une information supplémentaire pour le tri des scores. En effet, sachant le nombre de points qu'il faut trouver par sous-image, il n'est pas utile de trier l'intégralité de la matrice des scores mais plutôt d'en extraire uniquement le nombre de scores nécessaires. Ainsi, pour chaque sous-image, la matrice de score est parcourue une et une seule fois et le tableau à trier est de taille négligeable. Pour une image de 480 lignes pour 640 colonnes, 100 points sont recherchés avec une division de l'image en sous-ensemble de 8 lignes et 8 colonnes. Cette subdivision fait rechercher 2 points par sous-image de 4800 points et renvoie 128 points potentiels sur toute l'image. Une dernière phase de tri sur ce tableau permet d'obtenir le résultat présenté en figure 3.1c. De plus, la recherche d'un nombre de points faible peut faire

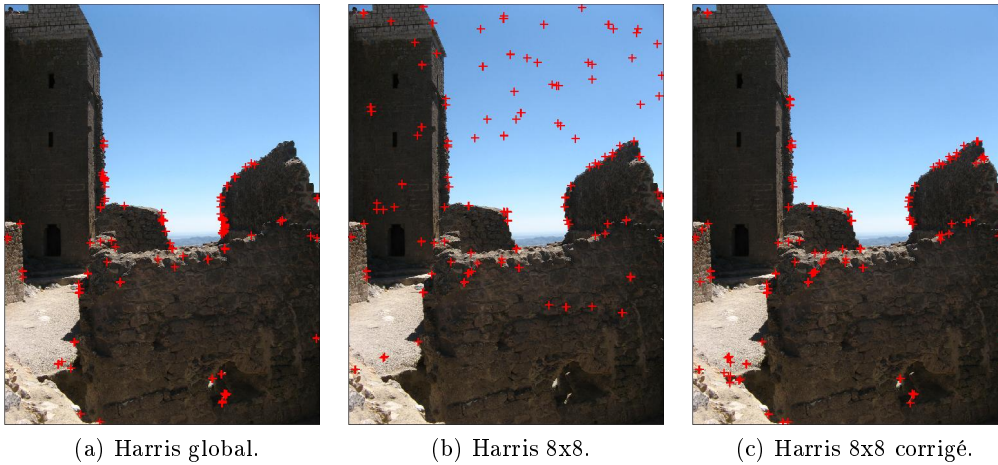


FIGURE 3.1 : Versions de l'extracteur de Harris (globale, découpée et découpée-corrigée) à la recherche de 200 points.

chuter la qualité d'un extracteur de Harris, car les points de fort gradient peuvent se trouver trop près les uns des autres et détruire ainsi le reste de l'information. Dans ce cas, l'exécution sur les sous-images résout ce problème comme le montre les figures 3.2a et 3.2b

Il est à noter qu'une étude préliminaire a été réalisée sur le principe d'un histogramme avec une dynamique suffisante pour étaler les scores correctement. La matrice de scores de Harris est reconstituée en intégralité sans effectuer de tri et est normalisée sur une dynamique définie (de 0 à 65535 par exemple). On construit un



FIGURE 3.2 : Les deux versions de l'extracteur de Harris pour un faible nombre de points (100 points).

tableau de la même dynamique et pour chaque pixel, la case du tableau dont l'indice est égale au score du pixel en question est incrémentée de 1. Une fois la matrice des scores traitée, on parcourt l'histogramme dans le sens inverse de la dynamique en sommant le nombre de chaque case parcourue. Dès que le nombre obtenu dépasse le nombre de points cherchés, on a trouvé le seuil qui permet de sélectionner, dans la matrice de score, les points à conserver. Cette technique donne des résultats équivalents à la méthode précédente en étant bien plus performante qu'une méthode de calcul de seuil par dichotomie. C'est le nombre de passages sur la matrice des scores qui a fait préférer la méthode de tri des résultats des sous-images à celle-ci.

Dès lors que l'on dépasse un point à rechercher par sous-image, on s'expose à la possibilité d'avoir un groupe de points qui correspondent à la même source d'information. Un filtre est appliqué pour éliminer ce cas. De même, il est possible que l'opérateur de Harris extrait des points qui correspondent à des points déjà répertoriés dans le vecteur d'état. C'est pourquoi nous appliquons le même filtrage afin qu'il ne reste au final que les points novateurs. Il n'est pas impossible qu'au terme des filtrages successifs il ne reste plus de points de Harris. Dès lors, aucun point ne sera ajouté malgré la demande induite par le manque d'observation. En d'autres termes, le procédé conclut qu'il n'y a pas de points d'intérêt dans l'image autres que ceux déjà présents dans le vecteur d'état ou dans la carte de référence.

À présent que nous avons un ensemble de points 2D dans l'image, nous allons aborder la phase de création des informations 3D correspondantes à chacun. Cette étape, identique pour chaque point, se décompose en deux parties :

- la création du point 3D fictif en lui-même, qui servira à initialiser le processus itératif ;
- la création de l'incertitude associée au point, soit la matrice de variance-covariance correspondante.

3.2.2 Création du point 3D

Suite à l'extraction des points de Harris ayant résisté aux diverses étapes de filtrage, il faut à présent créer un point 3D correspondant à chacun. Nous allons aborder ici la méthode traditionnelle (au travers du filtre de Kalman) et nous montrerons qu'elle n'est pas représentative de la réalité. Par conséquent, nous développerons de nouvelles approches pour finalement les comparer et faire notre choix. Au terme de cette section, nous serons capable, pour chaque point de Harris, d'initialiser un point 3D et de lui fournir une matrice d'incertitude cohérente.

Rappelons que le modèle d'observation ne permet pas d'observer la profondeur, il ne donne que la direction. Compte tenu du type d'information, la méthode par les angles vient naturellement comme le suggère la figure 3.3.

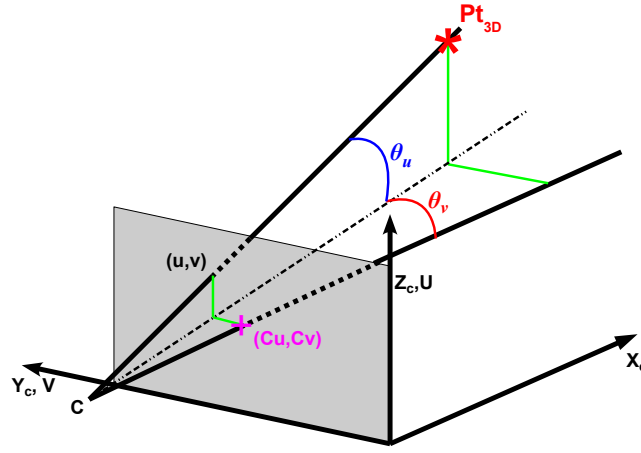


FIGURE 3.3 : Un point 3D peut se définir par deux angles (θ_u et θ_v) et une distance (coordonnées sphériques).

Spontanément, étant données les rotations \mathbf{R}_u et \mathbf{R}_v respectivement définies comme de fonctions de $u - C_u$ et $v - C_v$, on peut penser à créer le point fictif \vec{P}_{3D} sur le rayon de vue du point observé et de le déplacer sur le rayon de vue de l'observation à l'aide des rotations. Cependant, cette méthode signifie que l'on arrive au même résultat quel que soit l'ordre des rotations. Cette pensée peut être traduite par la relation suivante :

$$\mathbf{R}_u \mathbf{R}_v \vec{P}_{3D} = \mathbf{R}_v \mathbf{R}_u \vec{P}_{3D}$$

L'espace des matrices étant anti-commutatif pour des matrices non-symétriques, cette relation prouve que cette pensée intuitive est erronée. Restons dans une approche géométrique de la scène et constatons que l'on peut aussi écrire les équations à une profondeur X_0 définie (voir la figure 3.4 et l'annexe D). Cette représentation est cohérente et permet d'obtenir les coordonnées 3D du point qui initialisera le processus itératif au sein du filtre de Kalman. Le point créé dans le repère du capteur est

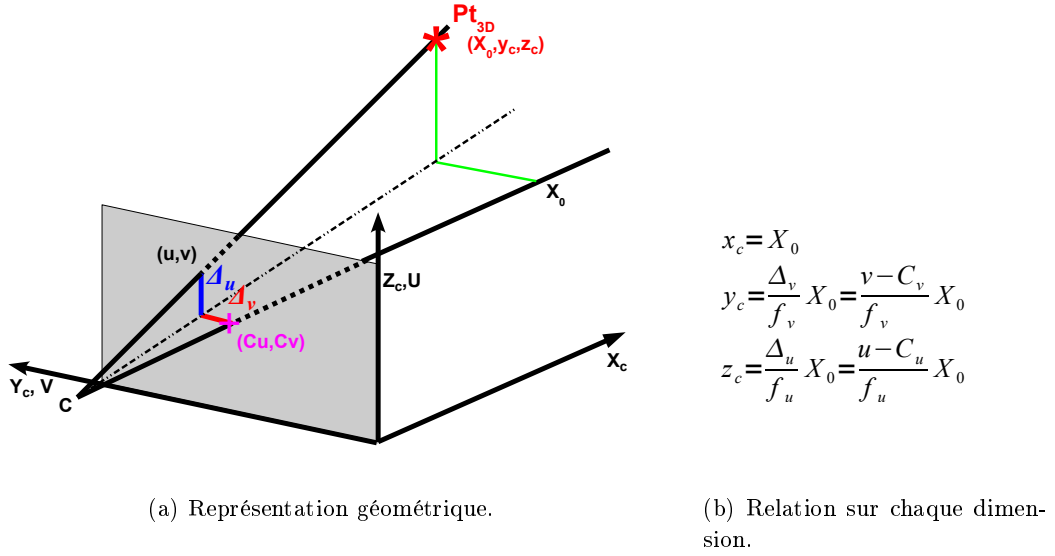


FIGURE 3.4 : Un point 3D peut se définir par trois distances (coordonnées cartésiennes). **3.4a** donne une illustration du problème en représentation cartésienne tandis que **3.4b** donne les relations permettant de construire un point 3D.

déplacé au repère véhicule puis au repère monde en lui appliquant successivement les rotations et translations correspondantes :

$$\vec{P}_{3D}^m = \mathbf{R}_{(v \rightarrow c)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}_{3D} - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow c)}) \quad (3.1)$$

3.2.3 Création de la matrice d'incertitude associée

Une fois le point 3D créé, il faut lui associer une incertitude qui soit cohérente avec l'observation. D'un point de vue purement statistique, cette incertitude peut être créée à l'aide du filtre de Kalman. Cependant, la représentation physique de l'incertitude générée n'est pas forcément cohérente avec la réalité. Par exemple, l'incertitude pourrait fournir, en tenant compte aussi de l'incertitude de l'observateur, une probabilité non négligeable que le point soit derrière l'observateur, cela étant impossible puisqu'il est observé ! Le fait que la création de l'incertitude ne soit pas contrôlée présente un inconvénient dans sa représentation et dans son utilisation future. Cependant, vouloir contrôler la création de l'incertitude amène à faire un nouveau choix :

- souhaite-t-on que la projection de l'ellipsoïde par les jacobiennes corresponde à l'incertitude dans l'image ?
- souhaite-t-on que la projection des points de l'ellipsoïde corresponde à l'incertitude dans l'image ?

Finalement, ces points vont être abordés successivement pour finalement choisir une méthode la plus cohérente possible.

3.2.3.1 Estimation de l'incertitude initiale d'un point par le filtre de Kalman

L'initialisation par la méthode du filtre de Kalman est la méthode classique. Elle consiste à faire une mise à jour de Kalman avec une innovation nulle. Le point 3D $\vec{P}t_{3D} = (x, y, z)$ est inséré dans le vecteur d'état à la suite des autres. La matrice d'incertitude est définie pour une incertitude sur l'axe de vue par :

$$\mathbf{P}_{Pt_{3D}} = \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{pmatrix} \quad (3.2)$$

Avec $\sigma_x = \sigma_y = \sigma_z \gg \sqrt{x^2 + y^2 + z^2}$ les variances suivant chaque axe. La prédiction de la position du point $\vec{P}t_{3D}$ construite à partir de l'observation correspond donc parfaitement avec cette dernière. Ainsi, l'innovation étant nulle, la mise à jour est nulle :

$$\begin{aligned} \vec{X}_{k+1|k+1} &= \vec{X}_{k+1|k} + \mathbf{G}(\text{innovation}) \\ &= \vec{X}_{k+1|k} + \mathbf{G} \times 0 \\ &= \vec{X}_{k+1|k} \end{aligned}$$

Cependant, la covariance sera réduite car la covariance de l'innovation est non nulle et vaut $\mathbf{H}\mathbf{P}_{k+1|k}\mathbf{H}^T + \mathbf{R}_0$ où $\mathbf{P}_{k+1|k}$ est la matrice d'incertitude de l'état après la phase de prédiction et \mathbf{R}_0 est la matrice de variance-covariance du bruit de mesure. Le résultat de la réduction d'incertitude est montré par la figure 3.5 (les incertitudes à 1σ et à 3σ sont représentées). On voit que cette incertitude, bien que correcte d'un point de vue statistique, ne traduit pas correctement l'aspect physique du processus. Les deux méthodes suivantes vont partir des caractéristiques observables pour essayer de déterminer un ellipsoïde d'incertitude plus cohérent avec la situation.

3.2.3.2 Estimation de l'incertitude initiale d'un point par les jacobienues du modèle d'observation

Une façon de contrôler la construction de l'incertitude est de vouloir, par exemple, que sa projection par les jacobienues du modèle d'observation corresponde à l'erreur d'observation. Contrairement au cas de la création du point, nous allons dans un premier temps montrer que l'on peut trouver les angles qui permettent de déplacer un ellipsoïde du rayon de vue principal sur le rayon de vue de l'observation. Dans un deuxième temps, nous verrons comment construire l'ellipsoïde qui nous intéresse.

Tout d'abord, plaçons-nous dans le repère associé au capteur. L'expression du point 3D $\vec{P}t_{3D}^m$ est $\vec{P}t_{3D}$. L'expression de l'homothétie qui place un point du rayon de vue du point principal sur celui défini par $\vec{P}t_{3D}$ est formulée ainsi :

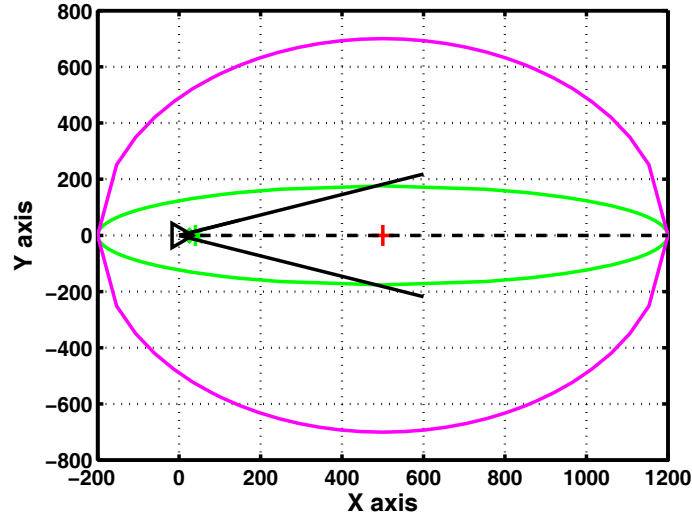


FIGURE 3.5 : Un point du monde (étoile verte) est observé depuis une position de l'observateur (triangle noir). L'initialisation à distance fixe conduit à la création du point fictif (croix rouge). Une incertitude initiale exagérée (ellipse rouge) est alors mise à jour avec l'observation pour fournir l'incertitude d'initialisation (ellipse verte). Bien que l'incertitude de l'observateur soit nulle, on constate la possibilité que le point soit situé derrière lui. Cela ne traduit pas correctement la physique de l'environnement.

$$\mathbf{R}_{Z_c}(\phi)\mathbf{R}_{Y_c}(\theta) \begin{bmatrix} X_0 \\ 0 \\ 0 \end{bmatrix} k = \begin{bmatrix} \cos(\phi)\cos(\theta) & -\sin(\phi) & \cos(\phi)\sin(\theta) \\ \sin(\phi)\cos(\theta) & \cos(\phi) & \sin(\phi)\sin(\theta) \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} X_0 \\ 0 \\ 0 \end{bmatrix} k = \begin{bmatrix} X_0 \\ y \\ z \end{bmatrix} \quad (3.3)$$

où X_0 est la profondeur max d'initialisation et θ et ϕ sont respectivement les angles de rotation autour des axes Y_c et Z_c qui déplacent le rayon de vue du point principal sur le rayon de vue du point observé. Ainsi, on peut extraire les angles de rotations suivants :

$$\phi = \arctan\left(\frac{y}{X_0}\right) \quad (3.4)$$

$$\theta = \arctan\left(\frac{-\cos(\phi)z}{X_0}\right) \quad (3.5)$$

$$(3.6)$$

Une fois ces angles obtenus, l'ellipsoïde, dont la forme quadratique est identifiée à la matrice \mathbf{E} , est déplacé sur le rayon de vue de l'observation via la relation suivante :

$$\begin{aligned} \mathbf{R}_c &= \mathbf{R}_z(\phi)\mathbf{R}_y(\theta) \\ \mathbf{E}_c &= \mathbf{R}_c\mathbf{E}\mathbf{R}_c^T \end{aligned} \quad (3.7)$$

A présent que les angles qui permettent de déplacer un ellipsoïde du rayon de vue principal au rayon de vue guidé par l'observation ont été déterminés, nous pouvons nous ramener au cas canonique d'ellipsoïde sur le rayon de vue principal, à savoir l'axe X_c . Dans ce qui suit, nous allons voir comment créer l'ellipsoïde en question.

Construire un ellipsoïde sur le rayon de vue n'est pas une tâche complexe si l'on connaît la longueur de ses axes principaux. Cependant, cela nous amène à nous poser les questions suivantes :

- Est-il vrai que cette incertitude générée sur le rayon de vue principal puis déplacée est bien cohérente avec ce que l'on souhaite construire ?
- Deux incertitudes définies en des lieux différents dans l'image sont-elles égales en tout point ?
- Déplacer une incertitude depuis le centre vers un point de l'image est-il cohérent en première approximation ?
- Comment définir une incertitude au centre de l'image qui puisse être cohérente du point de vue de la construction d'un ellipsoïde 3D, avec l'incertitude associée à l'observation ?

Nous allons apporter les réponses à chacune de ces questions.

Dans un premier temps, nous allons montrer qu'une incertitude constante dans l'espace image ne conduit pas à une incertitude angulaire constante. Pour ce faire, nous allons partir de la réciproque à savoir qu'une incertitude angulaire constante ne conduit pas à une incertitude centrée sur la projection de l'estimation.

On rappelle que θ_v est l'angle que ferait un rayon de vue par rapport au rayon principal dans le plan horizontal. Si on appelle σ_{θ_v} l'écart-type angulaire, alors la figure 3.6 illustre le propos. Ainsi, on peut établir les relations suivantes :

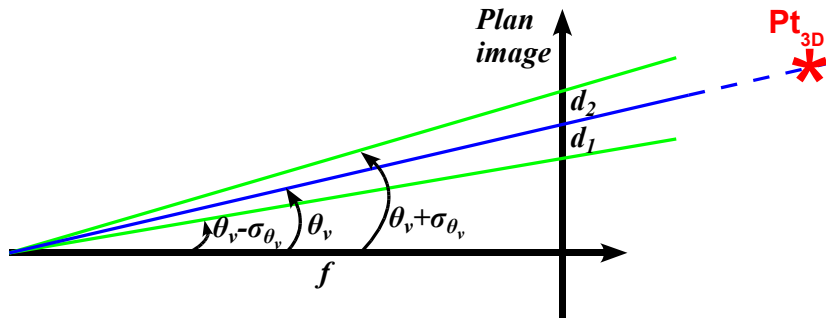


FIGURE 3.6 : Schématisation du problème de variation d'incertitude angulaire (f est la distance focale, θ_v l'angle que fait horizontalement le rayon de vue avec le rayon principal, σ_{θ_v} l'incertitude angulaire associée et d_1 et d_2 sont les distances à mesurer).

$$d_1 = f [\tan(\theta_v) - \tan(\theta_v - \sigma)] = f \frac{(1 + \tan(\theta_v)^2) \tan(\sigma_{\theta_v})}{1 + \tan(\theta_v) \tan(\sigma_{\theta_v})} \quad (3.8)$$

$$d_2 = f [\tan(\theta_v + \sigma) - \tan(\theta_v)] = f \frac{(1 + \tan(\theta_v)^2) \tan(\sigma_{\theta_v})}{1 - \tan(\theta_v) \tan(\sigma_{\theta_v})} \quad (3.9)$$

$$(3.10)$$

Ces relations mettent en évidence le fait que les écarts-types dans l'image varient en fonction de l'angle θ_v et que, pour un θ_v donné, elles ne sont pas égales entre elles. En prenant des valeurs courantes d'ouverture angulaire, de focale et d'écart-type en pixel, on justifie que cette variation n'est pas significative par les relations suivantes :

$$\begin{aligned} \text{focale} &= 1024 \text{ pixels} \\ \theta_v &= \pi/4 \text{ rd} \\ \sigma_{\theta_v} &= \text{atan}(4 \text{ pixels}/\text{focale}) \\ |d_2 - d_1| &\approx 0.05 \text{ pixels} \end{aligned} \quad (3.11)$$

Cependant, une incertitude angulaire constante peut faire doubler l'incertitude dans l'image en fonction de l'angle θ_v comme le montre la figure 3.7.

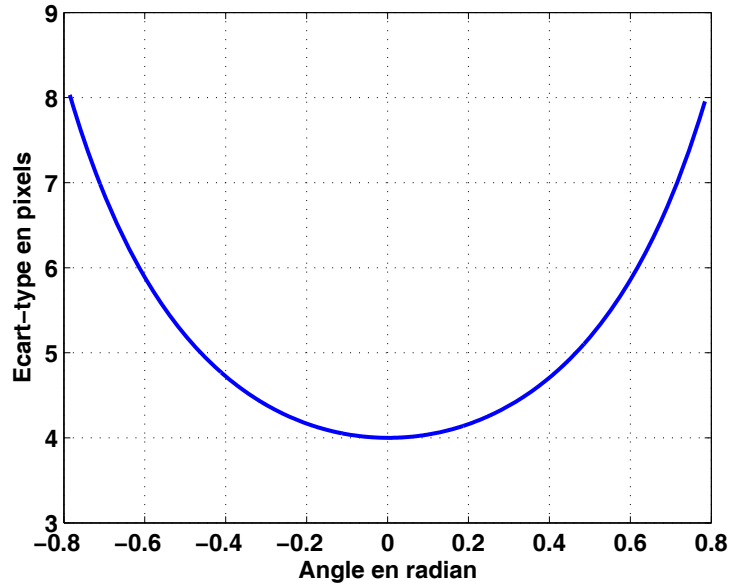


FIGURE 3.7 : Plus le rayon d'observation s'éloigne du rayon principal (plus l'angle θ_v est grand en valeur absolue), plus l'incertitude dans l'image croît pour une incertitude angulaire constante. Dans le cas d'une focale à 1024 pixels et d'une ouverture d'angle de $\pi/4$, l'incertitude peut même aller jusqu'à doubler. Ces relations, présentées pour l'axe V , sont valables sur l'axe U .

Que dire de cette brève analyse ? Revenons au problème de légitimité d'une incertitude fixe dans l'image. On a constaté qu'un angle d'incertitude constant conduit à

une variation de l'incertitude dans l'image et réciproquement. De plus, l'incertitude peut varier fortement au sein de l'image si l'on souhaite conserver une incertitude angulaire constante. L'erreur d'observation, quant à elle, se trouve au sein de l'image (définie en pixels sur chaque axe) et on vient de voir que sa représentation angulaire est variable selon la position du projeté. Une étude des relations 3.8 et 3.9 permet de définir quelle est l'incertitude autour du point central qui mène à une incertitude donnée en un point de l'image :

$$\begin{aligned}\sigma_{centre} &= \frac{f \times \sigma_{image}}{f \left(1 + \tan(\theta_v)^2\right) - \sigma_{image} \tan(\theta_v)} \\ &= \frac{f \times \sigma_{image}}{f \left(1 + \left(\frac{v-C_v}{f}\right)^2\right) - \sigma_{image} \left(\frac{v-C_v}{f}\right)}\end{aligned}\quad (3.12)$$

La figure 3.8 montre qu'effectivement, pour obtenir une incertitude de 4 pixels sur les bords de l'image, il faut définir une incertitude centrale moitié moins importante. Par

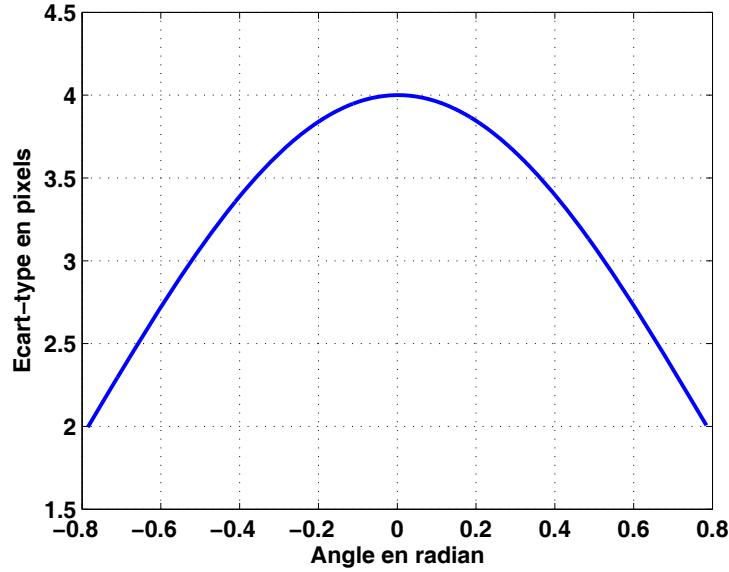


FIGURE 3.8 : Pour conserver un écart-type constant dans l'image (ici 4 pixels), la courbe, toujours sur l'axe V , donne les valeurs auxquelles il faut définir l'incertitude σ_v sur le projeté du point principal. Il en va de même pour les incertitudes sur l'axe U .

ailleurs, en admettant que ce qui vient d'être démontré dans le plan horizontal reste valable dans le plan vertical, on est en mesure de définir les incertitudes d'observation sur chaque axe qui guideront la construction de l'ellipsoïde.

Rappelons que l'on souhaite que cette incertitude soit équivalente à l'enveloppe de la projection par les jacobienness. Les calculs étant développés dans l'Annexe E.1,

on obtient les résultats suivants :

$$\begin{aligned}
 \rho &= \sqrt{X_0^2 + y^2 + z^2} \text{ et} \\
 \delta &= \text{distance minimale d'observation} \\
 \sigma_x &= \rho - \delta \\
 \sigma_y &= \rho \frac{\sigma_v}{f_v} \\
 \sigma_z &= \rho \frac{\sigma_u}{f_u}
 \end{aligned} \tag{3.13}$$

où X_0 est la profondeur d'initialisation fixée comme décrit dans l'annexe D. Sur le même exemple d'ellipsoïde que dans le cas d'une initialisation par le filtre de Kalman, la figure 3.9 donne la représentation d'une initialisation 2D. L'ellipsoïde créé

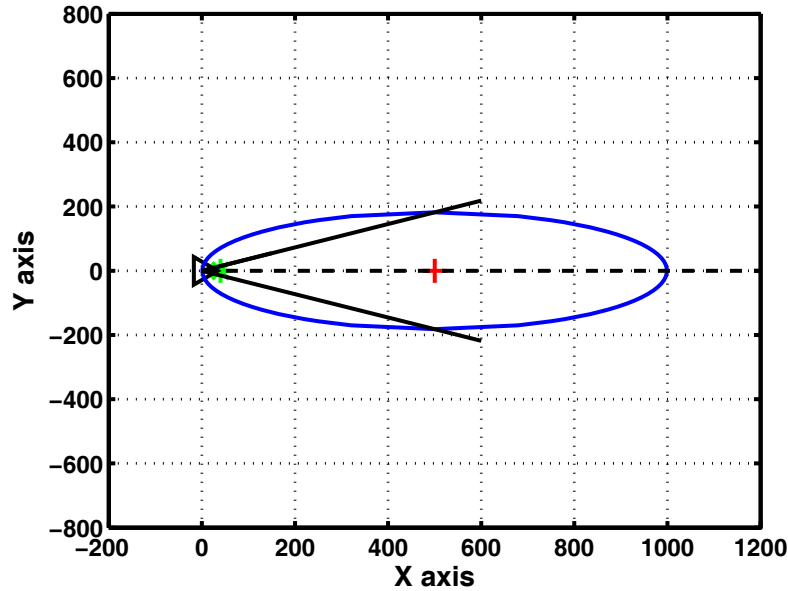


FIGURE 3.9 : Avec le même scénario que précédemment (3.5) et $\delta = 2$ unités, on constate que l'incertitude est plus grande que le cône d'observation sans pour autant permettre l'initialisation d'un point derrière l'observateur.

est ensuite propagé du repère capteur au repère monde, en tenant compte des incertitudes relatives des repères, afin de pouvoir être ajouté à la matrice d'incertitude du vecteur d'état.

3.2.3.3 Estimation de l'incertitude initiale d'un point par la projection de l'ellipsoïde d'incertitude

Comme dans le cas précédent, on se place dans le repère capteur avec le point sur l'axe optique. Cette fois-ci on cherche l'ellipsoïde dont la projection est contenue par l'incertitude définie dans l'image. Celui-ci correspond à celui axé sur le rayon

principal et tangent au cône d'observation de sommet le point principal et de base l'ellipse d'incertitude d'observation dans l'image. La figure 3.10 montre la représentation de la situation. Les calculs étant développés dans l'Annexe E.2, on obtient

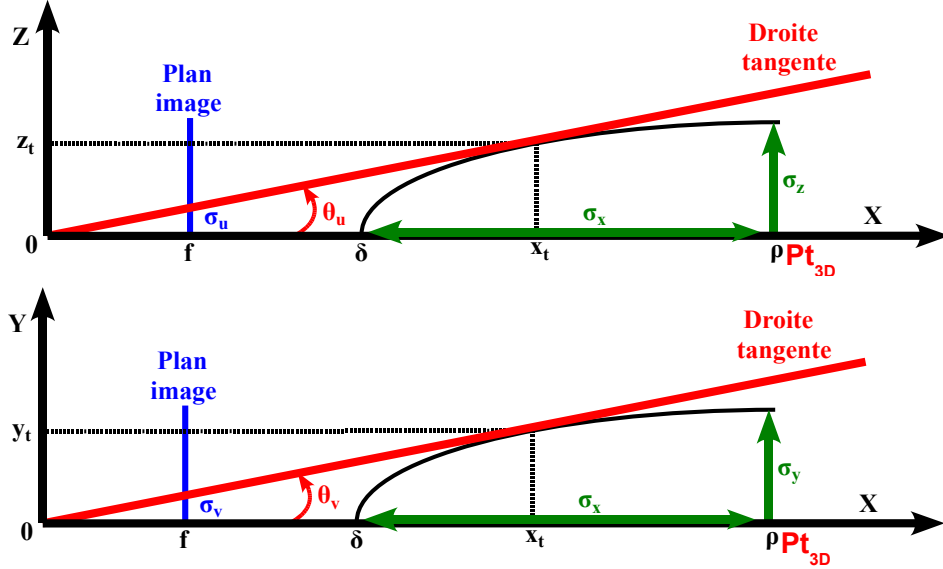


FIGURE 3.10 : L'ellipsoïde nominal étant défini sur l'axe \vec{X} , sans rotation, la figure illustre la projection de celui-ci dans les plans Oxz (en haut) et Oxy (en bas). S'exprime alors la notion de cône tangent défini dans les deux vues par la droite tangente rouge.

les résultats suivants :

$$\begin{aligned}
 \rho &= \sqrt{X_0^2 + y^2 + z^2} \\
 \delta &= \text{distance minimale d'observation} \\
 \sigma_x &= \rho - \delta \\
 \sigma_y &= \frac{\sigma_v}{f} \sqrt{\rho^2 - \sigma_x^2} \\
 \sigma_z &= \frac{\sigma_u}{f} \sqrt{\rho^2 - \sigma_x^2}
 \end{aligned} \tag{3.14}$$

Sur le même exemple d'ellipsoïde que dans le cas d'une initialisation par le filtre de Kalman, la figure 3.11 donne la représentation d'une initialisation 2D. L'incertitude créée est ensuite propagée, comme précédemment, du repère capteur au repère monde afin de pouvoir être ajoutée à l'incertitude du vecteur d'état.

3.2.3.4 Comparaison et choix

Nous venons de voir trois façons de créer un ellipsoïde d'incertitude à partir d'une même observation. La figure 3.12 offre une vue superposée des résultats de chaque méthode. La difficulté à lier l'incertitude générée par le filtre de Kalman avec

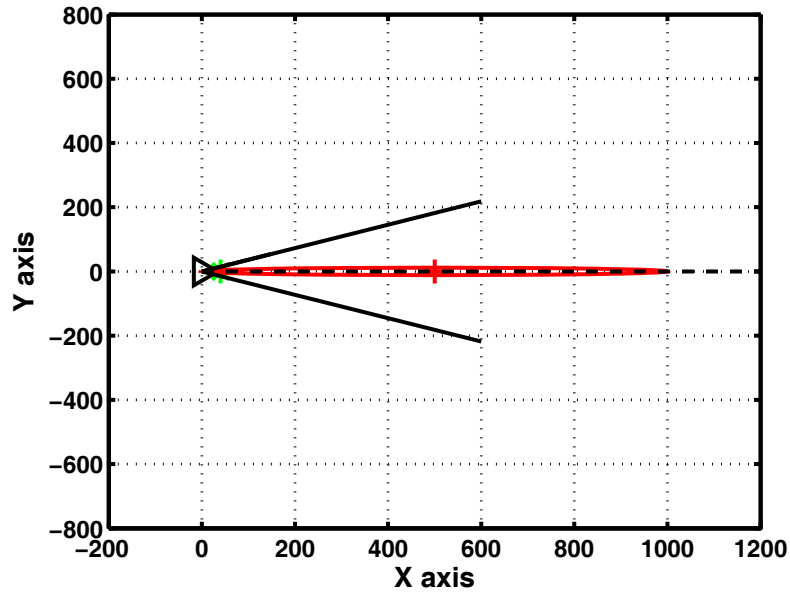


FIGURE 3.11 : Toujours avec le même scénario que précédemment (3.5) et $\delta = 2$ unités, on constate cette fois-ci que la méthode est plus restrictive.

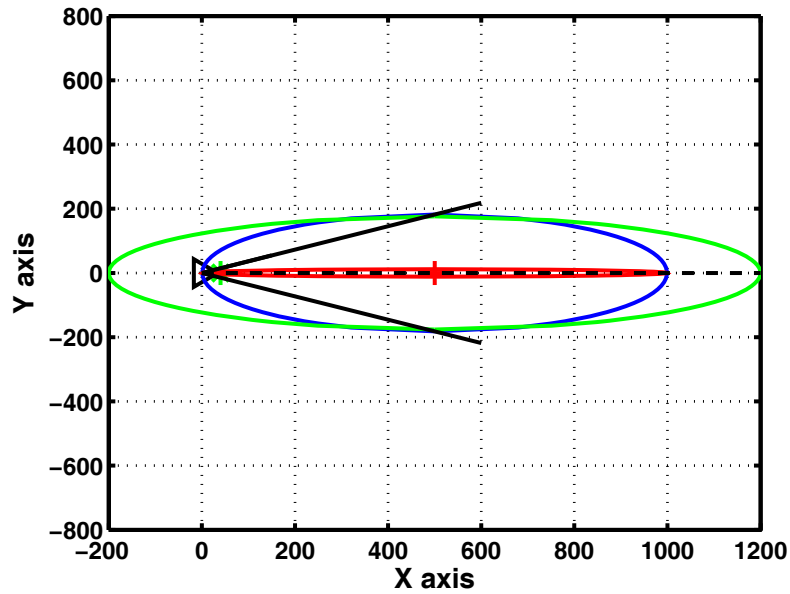


FIGURE 3.12 : La superposition des résultats permet de prendre conscience des avantages et des inconvénients de chaque méthode.

une interprétation physique cohérente n'encourage pas à sélectionner cette méthode. De plus, comme énoncé précédemment, l'incertitude dans l'image ne se projette pas sous une forme gaussienne dans le monde mais sous la forme d'un cône comme le

montre la figure 3.13. Cependant toute notre approche se fonde sur le principe de

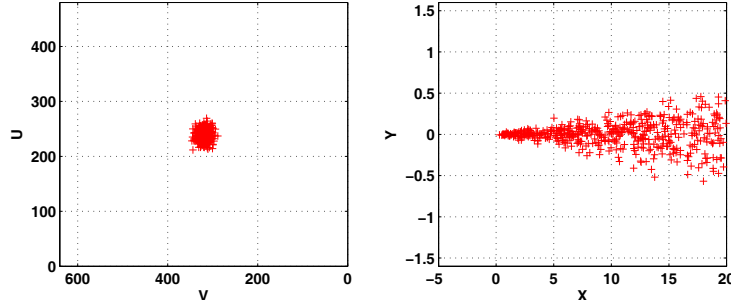


FIGURE 3.13 : A gauche, on définit des points dans une incertitude ($\sigma_u = \sigma_v = 10$ pixels). En affectant à chaque point une profondeur aléatoire (ici entre 1 m et 20 m), on peut reconstruire son homologue 3D. A droite, on visualise alors l'aspect non gaussien de l'incertitude représentée par l'enveloppe du nuage de point.

projection. Dans le chapitre précédent, nous avons vu comment calculer l'incertitude image associée à une incertitude 3D. En prenant le problème à l'envers, l'incertitude 3D correspondant à une incertitude image n'est autre que celle définie par la méthode des tangentes. Cette logique nous a fait sélectionner la dernière méthode présentée, en accord avec notre calcul de projection. Ainsi, nous avons établi une bijection entre les incertitudes 2D dans l'image et 3D dans le monde, sans laisser s'exprimer la non-linéarité du modèle d'observation.

3.3 Prédiction

Cette étape regroupe, comme nous l'avons vu lors du chapitre 2, deux phases à savoir la prédiction de la pose du véhicule via le modèle d'évolution et la prédiction de l'observation des points de chacune des cartes. C'est précisément dans cette étape que nous allons devoir introduire une nouvelle carte, construite par le véhicule tandis qu'il quitte la trajectoire de référence. Cette carte SLAM ne peut pas être mise directement en relation avec la carte de référence. En effet, la dérive du SLAM (que l'on nomme biais ou perte d'intégrité) doit être prise en compte. L'introduction et la prise en compte du biais dans les équations amènent une nouvelle fonction d'observation que nous allons présenter. En effet, l'observation d'un point de la carte de référence ne conduit pas aux mêmes équations que celles d'un point de la carte SLAM.

L'étape de prédiction de la pose du véhicule est la même que celle développée dans 2.3.1 et ne sera pas détaillée ici. Il en est de même pour l'observation des points de la carte SLAM, à savoir ceux engendrés par le processus SLAM. L'expression du biais se retrouve au niveau de l'observation des points de la carte de référence. Nous appellerons $\vec{B}_X = (b_x, b_y, b_z, b_{\alpha_x}, b_{\alpha_y}, b_{\alpha_z})^T$ les composantes du biais définissant

une translation \vec{T}_B et une rotation \mathbf{R}_B entre la carte SLAM et la carte de référence :

$$\vec{T}_B = (b_x, b_y, b_z)^T \quad (3.15)$$

$$\mathbf{R}_B = \mathbf{Rot}_z(b_{\alpha_z}) \mathbf{Rot}_y(b_{\alpha_y}) \mathbf{Rot}_x(b_{\alpha_x}) \quad (3.16)$$

Durant la phase d'évolution, on suppose que le biais évolue suivant un modèle *RandomWalk*² qui se traduit par les équations suivantes :

$$\vec{B}_{X_{k+1|k}} = \vec{B}_{X_{k|k}} \quad (3.17)$$

$$\mathbf{P}_{B_{k+1|k}} = \mathbf{P}_{B_{k|k}} + \mathbf{Q}_B \quad (3.18)$$

où \mathbf{Q}_B est la matrice diagonale de variance-covariance du bruit d'évolution.

L'observation des points $\vec{P}t$ de la carte SLAM se fait, sans prise en compte du biais, comme présentée dans la section 2.3.2. En ce qui concerne le calcul de l'observation des points $\vec{P}t'$ de la carte de référence, il faut introduire la notion de biais. La fonction d'observation devient la suivante :

$$u' = \frac{\vec{K}_2 \mathbf{R}_{(v \rightarrow c)}^T (\mathbf{R}'_{(m \rightarrow v)} (\vec{P}t' - \vec{T}'_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow c)})}{\vec{K}_1 \mathbf{R}_{(v \rightarrow c)}^T (\mathbf{R}'_{(m \rightarrow v)} (\vec{P}t' - \vec{T}'_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow c)})} = \frac{Nu'}{De'} \quad (3.19)$$

$$v' = \frac{\vec{K}_3 \mathbf{R}_{(v \rightarrow c)}^T (\mathbf{R}'_{(m \rightarrow v)} (\vec{P}t' - \vec{T}'_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow c)})}{\vec{K}_1 \mathbf{R}_{(v \rightarrow c)}^T (\mathbf{R}'_{(m \rightarrow v)} (\vec{P}t' - \vec{T}'_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow c)})} = \frac{Nv'}{De'} \quad (3.20)$$

$$\begin{aligned} \text{avec } \vec{T}'_{(m \rightarrow v)} &= \vec{T}_{(m \rightarrow v)} + \vec{T}_B \\ \mathbf{R}'_{(m \rightarrow v)} &= \mathbf{Rot}_z(\alpha_z + b_{\alpha_z}) \mathbf{Rot}_y(\alpha_y + b_{\alpha_y}) \mathbf{Rot}_x(\alpha_x + b_{\alpha_x}) \end{aligned}$$

où (u', v') est le couple de coordonnées image de la projection par \mathbf{K} de $\vec{P}t' = (x', y', z')$ et \vec{K}_l désigne la $l^{\text{ème}}$ ligne de \mathbf{K} ($l \in \{1, 2, 3\}$).

L'introduction du biais dans l'équation d'observation impose le calcul des jacobiennes par rapport à celui-ci. On peut remarquer qu'il intervient de manière additive sur les paramètres de position et d'orientation du véhicule dans le monde.

2. *Randomwalk* : marche aléatoire ; équivalent à un modèle de bruit brownien

L'écriture matricielle permet encore une fois d'en avoir une représentation simple :

$$\begin{aligned}\frac{\partial u'}{\partial b_x} &= \frac{\partial u'}{\partial X_v} = -\frac{\partial u'}{\partial x'} = C'_u \mathbf{R}_{(v \rightarrow c)}^T \mathbf{R}_{(m \rightarrow v)}'^T \begin{pmatrix} -1 & 0 & 0 \end{pmatrix}^T \\ \frac{\partial u'}{\partial b_y} &= \frac{\partial u'}{\partial Y_v} = -\frac{\partial u'}{\partial y'} = C'_u \mathbf{R}_{(v \rightarrow c)}^T \mathbf{R}_{(m \rightarrow v)}'^T \begin{pmatrix} 0 & -1 & 0 \end{pmatrix}^T \\ \frac{\partial u'}{\partial b_z} &= \frac{\partial u'}{\partial Z_v} = -\frac{\partial u'}{\partial z'} = C'_u \mathbf{R}_{(v \rightarrow c)}^T \mathbf{R}_{(m \rightarrow v)}'^T \begin{pmatrix} 0 & 0 & -1 \end{pmatrix}^T\end{aligned}\quad (3.21)$$

$$\begin{aligned}\frac{\partial u'}{\partial b_{\alpha_x}} &= \frac{\partial u'}{\partial \alpha_x} = C'_u \mathbf{R}_{(v \rightarrow c)}^T \left(\frac{\partial \mathbf{R}'_{(m \rightarrow v)}}{\partial \alpha_x} \right)^T \left(\vec{P}_{t'} - \vec{T}'_{(m \rightarrow v)} \right) \\ \frac{\partial u'}{\partial b_{\alpha_y}} &= \frac{\partial u'}{\partial \alpha_y} = C'_u \mathbf{R}_{(v \rightarrow c)}^T \left(\frac{\partial \mathbf{R}'_{(m \rightarrow v)}}{\partial \alpha_y} \right)^T \left(\vec{P}_{t'} - \vec{T}'_{(m \rightarrow v)} \right) \\ \frac{\partial u'}{\partial b_{\alpha_z}} &= \frac{\partial u'}{\partial \alpha_z} = C'_u \mathbf{R}_{(v \rightarrow c)}^T \left(\frac{\partial \mathbf{R}'_{(m \rightarrow v)}}{\partial \alpha_z} \right)^T \left(\vec{P}_{t'} - \vec{T}'_{(m \rightarrow v)} \right)\end{aligned}\quad (3.22)$$

Les jacobienness de v' selon les paramètres du biais s'expriment de la même manière que celle de u' en remplaçant $C'_u = \frac{De' \vec{K}_2 - Nu' \vec{K}_1}{De'^2}$ par $C'_v = \frac{De' \vec{K}_3 - Nv' \vec{K}_1}{De'^2}$ et $De' = \vec{K}_1 \mathbf{R}_{(v \rightarrow c)}^T (\mathbf{R}'_{(m \rightarrow v)} (\vec{P}_{t'} - \vec{T}'_{(m \rightarrow v)}) - \vec{T}'_{(v \rightarrow c)})$.

Les nouvelles équations d'observation 3.19 et 3.20 permettent de maintenir la cohérence entre les deux types de cartes. Les jacobienness calculées auront leur importance dans la mise à jour de la position du véhicule suite à l'observation des points de la carte de référence. En effet, les relations 3.19 et 3.20 permettent de lier un point de la carte de référence, la position du véhicule ainsi que les paramètres du biais.

Dans les équations du SLAM, ce biais n'est pas observable (on ne peut pas dissocier le mouvement du véhicule d'un biais du SLAM). Par contre, ce problème sera résolu dès l'observation d'un point de la carte de référence.

3.4 Mise à jour

L'étape de mise à jour consiste à venir préciser les paramètres du vecteur d'état suite à l'observation d'un élément plus ou moins bien connu. Le vecteur d'état est à présent composé des 6 paramètres du véhicule, des 6 paramètres du biais ainsi que de 3 coordonnées pour chaque point de la carte du SLAM.

3.4.1 Mise à jour avec la carte de référence

Dans ce cas, on observe dans l'image un point de la carte de référence. Cette étape est relativement similaire à l'étape développée lors de l'algorithme de localisation (chapitre 2) en remplaçant la fonction d'observation \mathbf{h} par son homologue \mathbf{h}' qui tient compte du biais et dont les composante u' et v' ont été définies précédemment.

Exactement comme lors de la phase de mise à jour de l'algorithme de localisation, on vient mettre à jour les composants du vecteur d'état, en l'occurrence les paramètres du véhicule, les paramètres du biais et les coordonnées des points de la carte SLAM (liée à la position du véhicule par les covariances).

3.4.2 Mise à jour avec la carte SLAM

Dans ce cas, on observe dans l'image un point de la carte SLAM. Malgré le fait que la fonction d'observation lors de la prise en compte des points de la carte SLAM soit la même que celle développée pour la localisation (chapitre 2) il n'en reste pas moins une différence fondamentale : les points doivent cette fois-ci être mis à jour. L'écart entre la prédiction d'un point n'ayant pas encore convergé et son observation fait naître le problème de la non-linéarité soulevé dans le chapitre précédent. Dans le cas du SLAM, elle n'intervient pas seulement au niveau de la zone de recherche dans l'image, mais aussi dans le calcul du gain de Kalman. Ce point est crucial car une mauvaise mise à jour entraîne la non-observation du point dans les images successives d'où une suppression de celui-ci qui implique alors une nouvelle initialisation. L'initialisation est un processus à part du déroulement du filtre de Kalman qui impose un temps de calcul supplémentaire, cette divergence fera l'objet de la section 3.5.

3.5 Problème de linéarisation du modèle d'observation

Le filtre de Kalman est optimal dans le cas des fonctions linéaires. [Ristic 2004] montre qu'il le reste pour des fonctions non-linéaires, à condition de linéariser autour du point vrai. Dans le cas du SLAM, cela revient à linéariser autour du point vers lequel on cherche à converger. Il est bien évident qu'au moment de l'initialisation, cette information n'est pas disponible car inconnue. C'est pourquoi la linéarisation se fait, non pas autour du point vrai, mais autour du point estimé. Cette différence fait que le filtre de Kalman étendu (ou EKF) est sous-optimal.

Nous allons montrer que, dans une certaine mesure, l'effet de la non-linéarité est détectable et qu'il peut donc être réduit. Pour mieux percevoir et contrôler le phénomène, on se propose de traiter un cas plus simple à savoir bidimensionnel, avec une fonction d'observation du même type que celle développée dans le Visual SLAM (un quotient de fonctions rationnelles du premier degré). Cette configuration est abordée dans la section 3.5.1. Ensuite, l'application des résultats dans le cas du SLAM visuel 3D sera traité dans la section 3.6. Le cadre applicatif ainsi que le cas d'étude 2D ont fait l'objet des publications [Féraud 2011a] et [Féraud 2011b].

3.5.1 Cas d'étude 2D : $z = y/x$

Afin d'observer la non-linéarité du modèle d'observation, le cas de la fonction d'observation qui à (x, y) associe le réel y/x dans l'espace capteur est traité. Cette

fonction, un quotient de deux formes linéaires, est semblable à la fonction de projection d'une caméra, le tout ramené à l'espace 2D. De la même manière, elle ne donne pas d'information sur la profondeur mais sur la direction de visée. La figure 3.14 montre une représentation du scénario mis en place. La position de l'observateur est

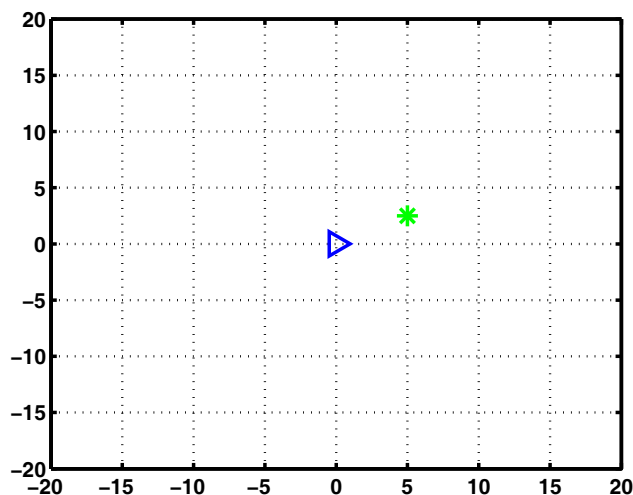


FIGURE 3.14 : Un amer 2D (étoile verte) est observé depuis une première position (triangle bleu).

supposée parfaitement connue ce qui ne laisse que la position de l'amer à découvrir : nous allons uniquement faire un travail de cartographie.

Suite à la première observation, un point fictif est créé. La figure 3.15 montre l'initialisation 2D avec une ellipse dans laquelle est inclus l'amer cherché. L'intégrité

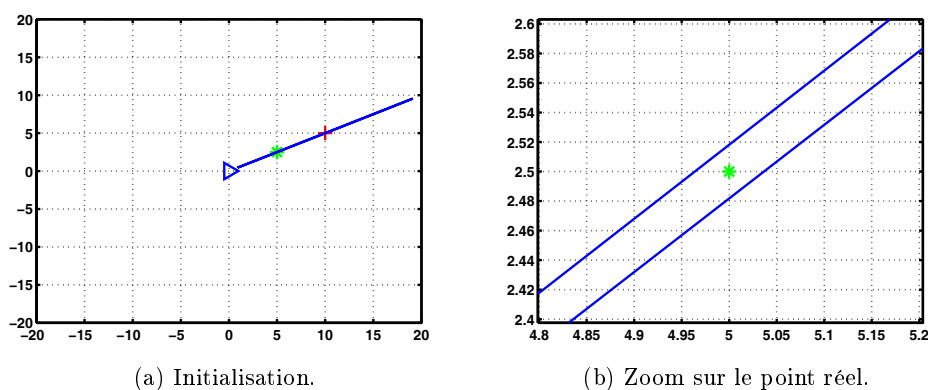


FIGURE 3.15 : **3.15a** : la première observation permet d'initialiser un point 2D (croix rouge) et son incertitude (ellipse bleue) correspondant à un écart-type de 2 pixels. **3.15b** : un zoom sur le point réel permet de visualiser l'intégrité du point réel vis-à-vis du point initialisé.

étant vérifiée, poursuivons le scénario avec le déplacement de l'observateur. On rappelle qu'à tout moment, sa position est parfaitement connue. La figure 3.16 présente l'observateur à sa nouvelle position. De là, il observe de nouveau l'amer. L'incerti-

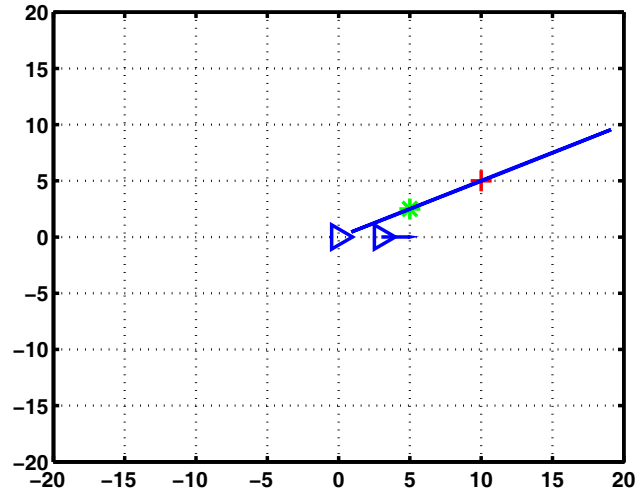


FIGURE 3.16 : Nouvelle position de l'observateur.

tude de l'observation est ici suffisamment faible pour être négligeable à l'échelle des distances du scénario. La mise à jour de la position estimée de l'amer suite à la nouvelle observation est donnée dans la figure 3.17. L'expression de la non-linéarité dans

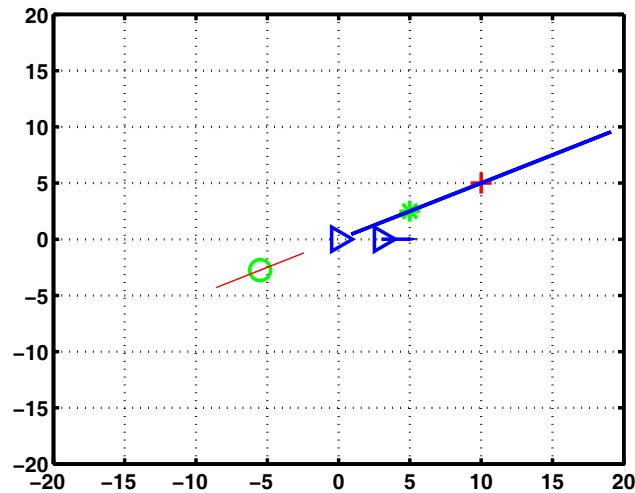


FIGURE 3.17 : Mise à jour traditionnelle par le filtre de Kalman (cercle noir associé à une incertitude verte). On constate alors la perte de l'intégrité qui traduit l'effet de la non-linéarité.

ce scénario place l'estimation de l'amer derrière la position de la première observation. Ce cas, identifié comme un cas de « profondeur négative » par [Martin 2008], a été traité dans le cas d'une paramétrisation en inverse de la profondeur.

Supposons un instant que l'observation ait eu une incertitude infinie : le filtre de Kalman n'aura pas modifié la position de l'amer estimée puisque cette incertitude intervient en inverse dans le calcul du gain de Kalman.

Il existe une méthode permettant de rendre les équations d'observation linéaires : chaque observation permet d'initialiser un point 3D et on met à jour un point 3D dans le monde avec un autre, tous deux correspondant à l'observation d'un même point. Dans ce cas, le système étant optimal, il aurait conduit à la solution présentée par la figure 3.18. Dans ce cas, deux observations suffisent pour trianguler une po-

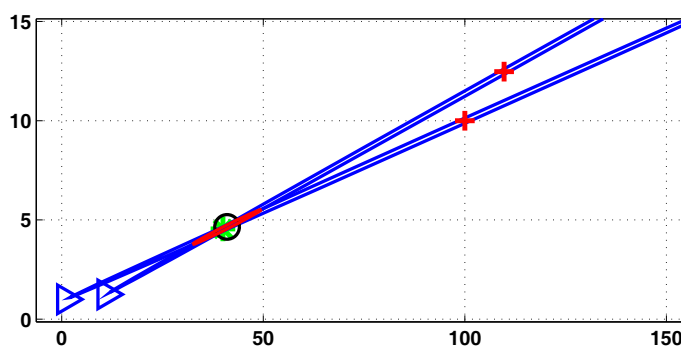


FIGURE 3.18 : Les deux observations d'un même point réel (étoile verte) conduisent chacune à l'initialisation d'un point 3D (croix rouge) et de leur incertitude associée (ellipse bleue). Dans le monde, la fusion de ces deux points est régie par un système linéaire. L'estimation, dans ce cas optimale, est donnée par le cercle noir et son incertitude, intègre vis-à-vis de la valeur vraie, est représentée par l'ellipse rouge.

sition mais cette méthode présente l'inconvénient de perdre la relation qu'a le point avec l'observateur lors de la mise à jour dans le monde. Par ailleurs, la contrainte épipolaire impose que le point se situe sur une droite portée par la projection de l'observation et la projection de la prédiction.

La figure 3.19 montre le résultat de mise à jour pour différentes erreurs d'observation, avec la fonction non-linéaire. Elle permet de constater que la mise à jour se fait en direction de l'observation. En rassemblant les diverses informations, on en arrive à l'énumération suivante :

- la projection de l'amer cherché se trouve sur une droite portée par l'observation et la projection de la prédiction (contrainte épipolaire) ;
- si l'observation est à erreur infinie, l'estimation n'est pas mise à jour ;
- si la fonction d'observation est linéaire et sans erreur, la projection de la mise à jour coïncide avec l'observation ;
- lorsque l'erreur d'observation décroît et dans le cas d'une fonction non-linéaire définie comme un quotient de deux formes linéaires du premier degré, alors la projection de la mise à jour se déplace de l'estimation vers l'observation.

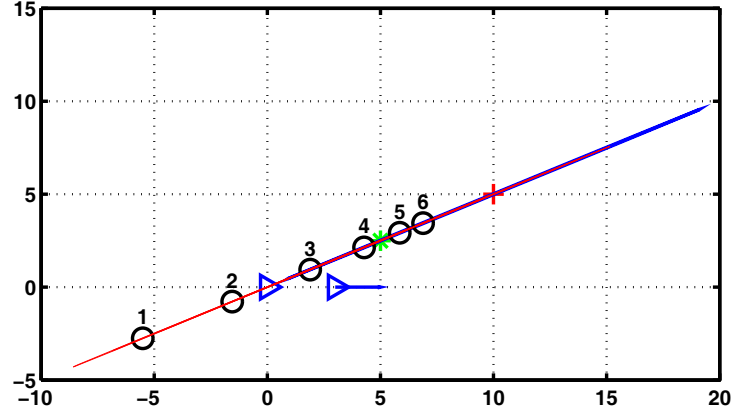


FIGURE 3.19 : La fonction d'observation est non-linéaire et de la forme y/x . Sur le scénario précédent, on effectue, suite à la seconde observation, différentes mises à jour avec différentes valeurs d'écart-type pour l'erreur d'observation (valeur indiquée au dessus de la mise à jour correspondante). On constate le déplacement linéaire de la mise à jour : plus l'incertitude d'observation est grande, plus la mise à jour est faible.

L'erreur d'observation étant définie sur \mathbb{R}^{*+} , alors la projection de la mise à jour doit se trouver sur le segment $[estimation, observation]$, *estimation* étant la projection de l'estimée du point 3D et *observation* étant le point image, projeté du point réel. Si elle n'appartient pas à cet intervalle, alors il faut limiter la mise à jour de telle sorte que sa projection coïncide avec la borne maximale, à savoir l'observation. Le gain de Kalman limite se calcule donc lorsque la projection de la mise à jour égale l'observation :

$$proj(\vec{X}_{(k+1|k+1)}) = observation \quad (3.23)$$

Cette relation est à la base du calcul de correction proposé qui a pour fonction de modifier le gain de Kalman dans la mise à jour.

3.5.2 Calcul de la correction

Une fonction d'observation non-linéaire génère des perturbations qui peuvent être décelées dans la mesure où cette fonction peut s'écrire comme un quotient de deux formes linéaires. On va, dans cette partie, présenter la méthode de calcul du correctif du gain de Kalman pour assurer que la projection de la mise à jour se trouve dans l'intervalle $[estimation, observation]$ défini précédemment.

La relation 3.23 va permettre de calculer quel devrait être le gain de Kalman pour que la projection de la mise à jour coïncide avec l'observation. Dans une application météorologique, [Palmer 2006] propose une solution en pondérant le gain de Kalman sans pour autant préciser un cadre applicatif ou un domaine de définition. Nous allons voir ici, comment définir ce coefficient et comment son écriture nous mène à son domaine de définition. Soit \mathbf{G}' le gain de Kalman recherché et défini

proportionnellement par rapport à \mathbf{G} gain de Kalman traditionnel :

$$\mathbf{G}' = r \times \mathbf{G} \quad (3.24)$$

r désigne donc le coefficient déterminant la proportion du gain de Kalman qu'il faut prendre (équivalent à un pourcentage de celui-ci). La figure 3.19 montre aussi qu'en modifiant l'équilibre entre l'incertitude de l'observation \mathbf{R}_o et l'incertitude de l'estimation \mathbf{P} , on influe sur le déplacement de la mise à jour. Si $\mathbf{R} \gg \mathbf{P}$, la mise à jour tend à rester sur l'estimation tandis que si $\mathbf{R} \ll \mathbf{P}$, la mise à jour se déplace vers l'observation. Il serait donc possible de réduire l'effet de la non-linéarité en déséquilibrant le rapport entre les deux matrices d'incertitude mais cela reste impropre. Nous allons donc montrer que le gain \mathbf{G}' ne modifie pas cet équilibre. Ce gain peut se développer à partir de l'écriture du gain de Kalman comme suit :

$$\begin{aligned} \mathbf{G}' &= r \times \mathbf{G} \\ &= r \times \mathbf{P}\mathbf{H}^T (\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}_o)^{-1} \\ &= \mathbf{P}\mathbf{H}^T \left(\mathbf{H} \frac{\mathbf{P}}{r} \mathbf{H}^T + \frac{\mathbf{R}_o}{r} \right)^{-1} \\ &= \mathbf{P}\mathbf{H}^T (\mathbf{H}\mathbf{P}'\mathbf{H}^T + \mathbf{R}'_o)^{-1} \end{aligned} \quad (3.25)$$

avec $\mathbf{P}' = \frac{\mathbf{P}}{r}$ et $\mathbf{R}'_o = \frac{\mathbf{R}_o}{r}$

La relation 3.25 prouve que l'introduction du coefficient multiplicateur r touche les incertitudes d'observation et d'estimation dans les mêmes proportions, garantissant l'équilibre calculé par le gain de Kalman. De plus, on peut déduire les bornes de considération de r . En effet, \mathbf{R}_o étant la meilleure connaissance que l'on ait de l'erreur du bruit de mesure, \mathbf{R}'_o , qui reste une erreur d'observation, ne peut être plus précise. La relation $\mathbf{R}'_o \geq \mathbf{R}_o$ conduit à l'inégalité $r \leq 1$. De plus, les matrices d'incertitudes sont semi-définies positives ce qui s'avère être une condition essentielle au bon déroulement du filtre de Kalman. Le cas de la matrice \mathbf{P} permet de calculer la borne inférieure de l'intervalle de définition de r . En effet, \mathbf{P} étant une matrice symétrique réelle, elle est par définition diagonalisable par le groupe orthogonal. La semi-définie positivité permet de dire que les valeurs propres de \mathbf{P} sont supérieures ou égales à 0. Elles traduisent, dans le cas 2D, le carré de la longueur de chacun des deux axes d'une ellipse canonique ; les matrices de passage étant la matrice de rotation et sa transposée qui donnent l'orientation à l'ellipse canonique. Par essence, ces valeurs propres ne peuvent être négatives. Par conséquent :

$$\begin{aligned} \mathbf{P}' &= \frac{\mathbf{P}}{r} \\ &= \mathbf{Rot} \frac{\mathbf{Diag}}{r} \mathbf{Rot}^{-1} = \mathbf{Rot} \frac{\mathbf{Diag}}{r} \mathbf{Rot}^T \\ \Rightarrow r &> 0 \end{aligned} \quad (3.26)$$

Ainsi, on trouve que r ne doit être pris en compte comme correctif du gain de Kalman uniquement s'il appartient à l'intervalle $]0 \ 1]$. A présent que le domaine

de définition est établi, le calcul de r se fait directement à partir des relations 3.23 et 3.24. De la même manière que dans le cas de la localisation et du SLAM, le modèle d'observation doit replacer le point dans le repère du capteur afin de lui affecter la fonction de projection. Aussi, soit \mathbf{R}_{2D} et \vec{T}_{2D} respectivement la rotation et la translation, définies à partir des 3 degrés de liberté de l'observateur, qui place l'observateur dans le repère monde. La fonction d'observation devient alors \mathbf{h}_{2D} telle que :

$$\mathbf{h}_{2D} : \vec{X}_{2D} \mapsto \frac{\begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{R}_{2D}(\vec{X}_{2D} - \vec{T}_{2D})}{\begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{R}_{2D}(\vec{X}_{2D} - \vec{T}_{2D})} \quad (3.27)$$

Si on note Δ l'innovation, alors le vecteur d'état mis à jour \vec{X}_{2D}^{upd} s'écrit quant à lui de la manière suivante :

$$\vec{X}_{2D}^{upd} = \vec{X}_{2D} + \mathbf{G}'\Delta \quad (3.28)$$

où \vec{X}_{2D} désigne le vecteur d'état qui se trouve être construit, dans le cas de la cartographie, comme la concaténation des coordonnées des points de la carte (dans le cas du filtre de Kalman utilisé, les points sont traités successivement). La relation 3.23 permet alors de trouver r :

$$\begin{aligned} obs &= \mathbf{h}_{2D}(\vec{X}_{2D}^{upd}) \\ &= \frac{\begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{R}_{2D}(\vec{X}_{2D}^{upd} - \vec{T}_{2D})}{\begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{R}_{2D}(\vec{X}_{2D}^{upd} - \vec{T}_{2D})} \\ &= \frac{\begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{R}_{2D}(\vec{X}_{2D} + \mathbf{G}'\Delta - \vec{T}_{2D})}{\begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{R}_{2D}(\vec{X}_{2D} + \mathbf{G}'\Delta - \vec{T}_{2D})} \text{ et } \mathbf{G}' = r \times \mathbf{G} \\ \Rightarrow r &= \frac{(obs - est) De}{(\begin{bmatrix} 0 & 1 \end{bmatrix} - obs \begin{bmatrix} 1 & 0 \end{bmatrix}) \mathbf{R}_{2D} \mathbf{G} \Delta} \end{aligned} \quad (3.29)$$

avec $De = x_{est} = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{R}_{2D}(\vec{X}_{2D} - \vec{T}_{2D})$ et les abréviations *obs* et *est* désignent respectivement l'observation et l'estimation sur la dimension étudiée (dans ce cas présent, la fonction d'observation est unidimensionnelle). En rassemblant les relations relatives au coefficient de pondération r , on peut, pour chaque point, vérifier si une correction est à appliquer :

$$r = \frac{(obs - est) De}{(\begin{bmatrix} 0 & 1 \end{bmatrix} - obs \begin{bmatrix} 1 & 0 \end{bmatrix}) \mathbf{R}_{2D} \mathbf{G} \Delta} \quad (3.30)$$

$$\mathbf{G}' = \begin{cases} r \times \mathbf{G} & \text{si } r \in]0, 1] \\ \mathbf{G} & \text{sinon} \end{cases} \quad (3.31)$$

3.5.3 Résultat de la correction

En reprenant l'exemple canonique bi-dimensionnel, la figure 3.20 montre l'application du correcteur de gain de Kalman. La valeur du coefficient de pondération est $r = 0.3225$, ce qui signifie que la mise à jour est limitée au tiers de ce qu'elle aurait été. La non-linéarité est d'autant plus forte que l'innovation est grande et

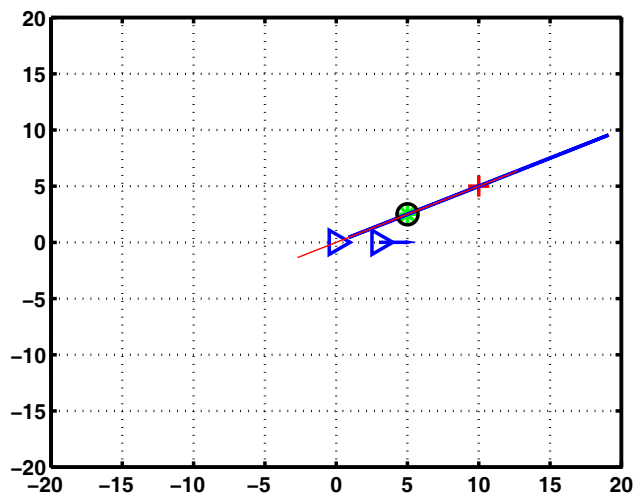


FIGURE 3.20 : La mise à jour corrigée (cercle noir) et son incertitude (ellipse rouge) sont intègres vis-à-vis de l'observation : l'estimation est cohérente et, en conséquence, son incertitude est moins réduite que dans le cas non-corrigé.

que l'incertitude d'observation est faible. De plus, le coefficient de pondération ne corrige pas uniquement la position estimée du point, qui est intégrée au gain de Kalman, mais également la mise à jour de l'incertitude qui est réduite en conséquence. Ceci explique que l'incertitude obtenue après correction soit plus grande que celle obtenue sans.

Finalement, cette correction n'est pas spécifique au cadre du SLAM visuel et se généralise pour toutes les fonctions non-linéaires s'exprimant sous la forme d'un quotient de deux formes linéaires du premier degré. L'hypothèse d'indépendance permet aussi de porter cette correction à un cadre multi-dimensionnel, donnant une correction applicable sur chaque dimension. C'est ce que nous allons voir en repassant dans le monde 3D avec une observation 2D.

3.6 Retour au Visual SLAM

Dans le cas du Visual SLAM, l'observation est une fonction possédant 2 champs. Sous l'hypothèse d'indépendance des coordonnées, on peut alors, de la même manière que dans le cas 2D, venir calculer un coefficient de pondération sur chacune d'elles. Etant entendu qu'il ne faut pas rajouter de l'information au travers de la correction, c'est le plus petit coefficient qui sera conservé, à la condition toutefois

qu'il appartienne au domaine de définition $]0, 1]$. La correction se fera donc ainsi :

$$r_u = \frac{(u_{obs} - u_{est})D_e}{(\mathbf{K}_2 - u_{obs}\mathbf{K}_1)\mathbf{R}_{(v \rightarrow c)}^T \mathbf{R}_{(m \rightarrow v)}^T \mathbf{Gpt}\Delta} \quad (3.32)$$

$$r_v = \frac{(v_{obs} - v_{est})D_e}{(\mathbf{K}_3 - v_{obs}\mathbf{K}_1)\mathbf{R}_{(v \rightarrow c)}^T \mathbf{R}_{(m \rightarrow v)}^T \mathbf{Gpt}\Delta} \quad (3.33)$$

$$r = \begin{cases} r_u & \text{si } 0 < r_u < r_v \text{ et } r_u < 1 \\ r_v & \text{si } 0 < r_v < r_u \text{ et } r_v < 1 \\ 1 & \text{sinon} \end{cases} \quad (3.34)$$

$$\mathbf{Gpt}' = \begin{cases} r \times \mathbf{Gpt} & \text{si } r \in]0, 1] \\ \mathbf{Gpt} & \text{sinon} \end{cases} \quad (3.35)$$

où \mathbf{Gpt} désigne la partie du gain de Kalman qui correspond à la mise à jour en regard du point en question (\vec{Pt}).

Avant de lancer la correction sur l'intégralité du parcours, nous allons étudier le cas d'un point 3D et de l'impact de la correction pour différentes observations définies dans l'image. Le scénario est simple et sa représentation est donnée par la figure 3.21. Depuis une première position, un rayon de vue sur lequel on place le

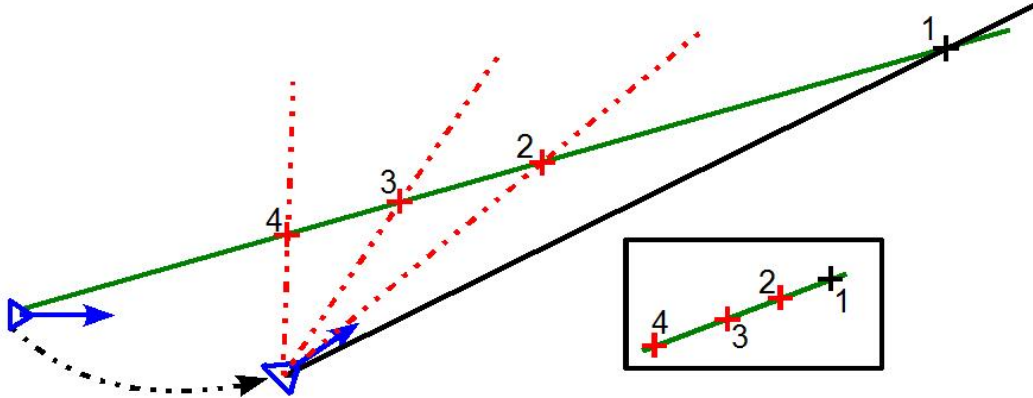
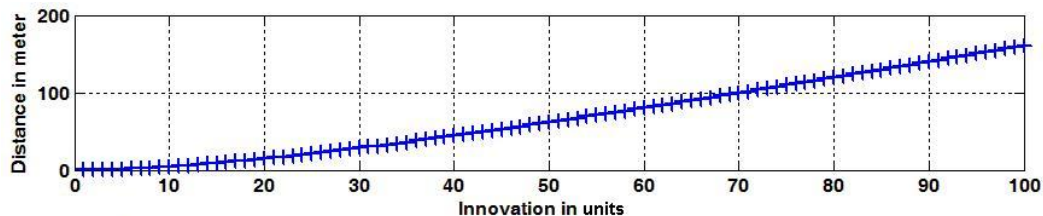


FIGURE 3.21 : Depuis une première position, on crée un point suffisamment loin (croix noire). Depuis une seconde position, on observe ce point ainsi que l'intégralité du rayon de vue (ligne verte) à l'aide d'un capteur de type caméra. Au sein de l'image (cadre noir en bas à droite), on définit, sur la projection du rayon de vue, une série d'observations (croix rouges). Par une simple triangulation, on trouve la position d'un point relatif à une observation sur le rayon de vue 3D.

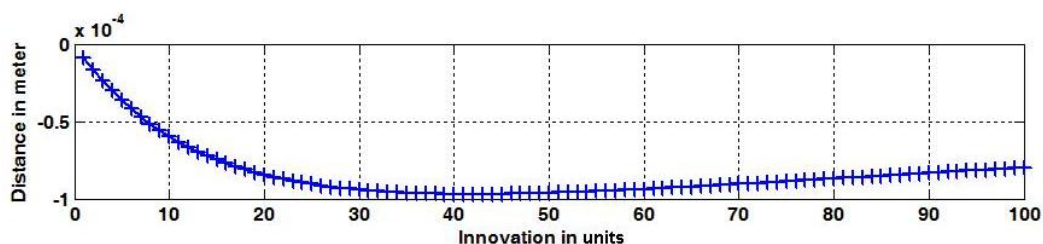
point fictif (labellisé 1 dans la figure 3.21) est défini. Depuis une deuxième position, on projette dans l'image le point fictif et le rayon de vue. A partir de là, nous allons contrôler l'innovation pour observer où se situe la mise à jour sur le rayon de vue initial.

Etant donné que nous sommes en simulation, il est donc possible de calculer l'intersection des rayons de vue issus de chaque observation. Par conséquent, la

position vraie du point 3D est directement observable par triangulation. En faisant croître l'innovation dans l'image (points 2, 3 et 4 dans la figure 3.21), on obtient les courbes de comparaison 3.22a et 3.22b. On observe alors encore une fois l'impact de



(a) Résultats sans correction.



(b) Résultats avec correction.

FIGURE 3.22 : 3.22a : la distance entre chaque mise à jour non corrigée et le vrai point obtenu par triangulation. 3.22b : la distance entre chaque mise à jour corrigée et le vrai point obtenu par triangulation.

la correction qui rend le système plus stable et moins divergent.

Remarquons qu'il n'est pas nécessaire que le point passe derrière l'observateur pour que la correction soit appliquée, comme c'est le cas dans [Martin 2008] (avec une paramétrisation du point en profondeur inverse). En des termes plus simples, il suffit que la projection du point mis à jour passe au delà de l'observation sur la droite épipolaire.

A présent, on pourrait lancer l'algorithme uniquement en mode SLAM sans carte de référence (ce qui implique la non prise en compte du biais) pour voir le résultat de la correction. L'algorithme va donc cartographier l'environnement et simultanément se localiser, comme le ferait le premier véhicule qui défriche le parcours et doit donc construire la carte qui sera utilisable par les véhicules suivants. Ce test est réalisé dans l'environnement d'expérimentation **PAVIN**³ situé à Clermont-Ferrand. La trajectoire acquise est représentée par la trace rouge dans la figure 3.23. Les résultats de ce test, avec et sans correction, sont fournis dans le tableau 3.2 et la figure 3.24b. On constate un nombre de points plus importants dans la carte finale dès lors que l'on utilise la correction du gain de Kalman. En effet, du fait de la disparition de la divergence, les points extraits des images sont observés plus longtemps. Par conséquent, cela explique le nombre d'initialisations (création de points 3D), plus important dans le cas non corrigé. De plus, la phase d'initialisation

3. Plateforme d'Auvergne pour les Véhicule INtelligent

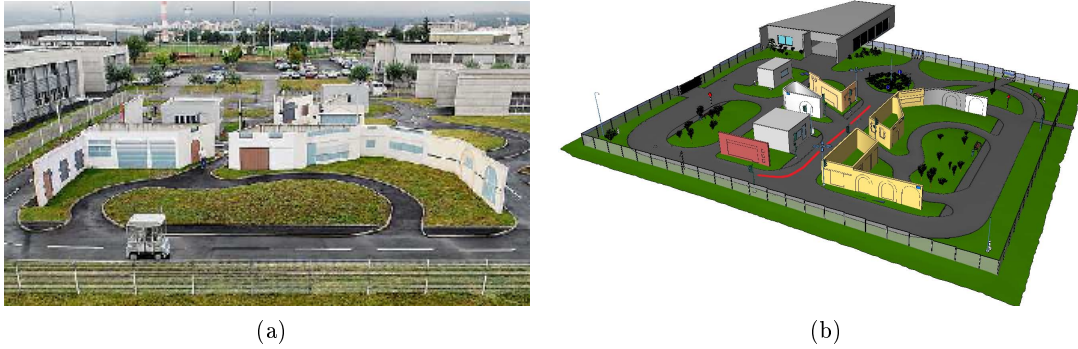


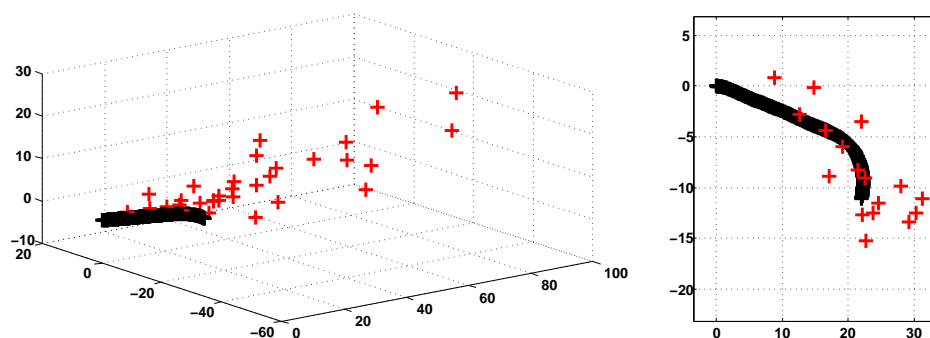
FIGURE 3.23 : **3.23a** : une vue réelle de l'environnement PAVIN en cours d'utilisation par un RobuCab. **3.23b** : une vue virtuelle de l'environnement sur laquelle est représentée la trajectoire d'étude (trace rouge).

	Avec correction	Sans correction
Nombre de points	75	32
Nombre d'initialisations	134	384
Nombre de divergences	0	439
Nombre de corrections détectées	264	591
Correction la plus importante	0.0024134	x

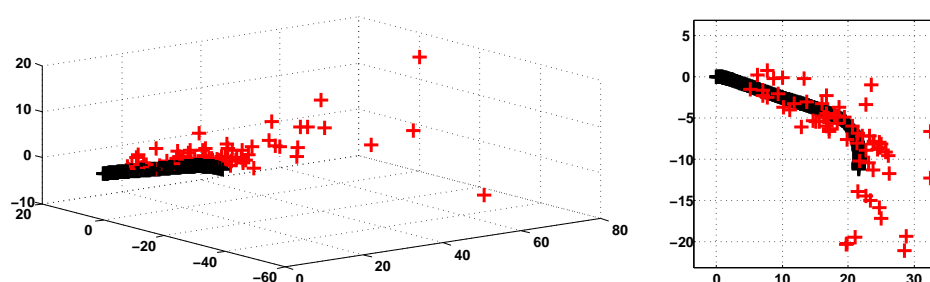
TABLE 3.2 : Tableau comparatif des résultats d'un SLAM visuel avec et sans correction du gain : la distance de travail est de 28 mètres, la vitesse reste inférieure à 1 m/s pour une cadence caméra de 10 Hz.

de points étant une phase supplémentaire et plus longue en temps de calcul, la version non-corrigée est plus lente : les temps n'ont pas été explicités ici étant donné qu'ils n'ont pas vraiment de sens avec un langage interprété tel que Matlab. Les divergences, quant à elle, sont définies dès lors qu'une mise à jour n'est plus intègre vis à vis de l'observation. Cela se traduit par le fait que l'observation courante n'est pas intègre avec la projection de la mise à jour qu'elle génère : c'est justement dans ce cas que la correction entre en jeu. Les figures 3.24a et 3.24b donnent entre autre, la disposition des points par rapport à la trajectoire. Il est à remarquer que, dans la version non-corrigée, peu de points proches ont été estimés correctement. Cela vient du fait que l'écart entre un point estimé et son observation est suffisamment important pour laisser s'exprimer la non-linéarité du modèle d'observation et faire diverger sa localisation.

Reste enfin une question : comment déterminer si un point, après un certain temps de non-observation, est un point qui doit être gardé ou si c'est un point qui n'a pas eu le temps de converger ? La méthode explicitée est celle qui a été utilisée pour obtenir les cartes de la figure 3.24.



(a) SLAM sans correction du gain de Kalman



(b) SLAM avec correction du gain de Kalman

FIGURE 3.24 : **3.24a** : Carte et trajectoire sans correction. **3.24b** : Carte et trajectoire avec correction. Dans chaque cas on trouve, à gauche, une vue 3D de la carte (croix rouges) et de la trajectoire (croix noires) et, à droite, un focus sur la trajectoire.

3.7 Conservation d'un point

Un point devient précis dès que son incertitude est suffisamment petite. Quantifier cette précision n'est pas si simple et la première idée qui consiste à seuiller les incertitudes ne s'avère pas forcément être la meilleure. [Joly 2010] donne une manière élégante de calculer le volume d'une zone d'incertitude 3D pour un écart-type donné. A partir de là, en se fixant un volume seuil, on peut déterminer quels sont les points à garder et quels sont ceux qui ne doivent pas l'être. Cependant cette méthode a un inconvénient : elle ne tient pas compte de la distance à laquelle se situe le point. En exemple, prenons 2 points, l'un situé à 1 mètre devant soi et l'autre à 100 mètres. Dire que l'on souhaite une contrainte volumique de 10 cm^3 n'a pas le même sens pour un point proche et pour un point loin. Il faudra beaucoup plus d'observations du point loin pour arriver à une telle contrainte. Cependant, un point à 100 mètres avec une incertitude de 10 m^3 s'avère être aussi utile qu'un point à 1 m avec une incertitude volumique de 10 cm^3 .

Une façon de tenir compte de la distance est de seuiller par une distance de

Mahalanobis. Malheureusement, il reste encore une faille dans cette approche : les dimensions peuvent se compenser. En effet, comme le montre la figure 3.25, il existe une infinité d'ellipsoïdes pour une distance de Mahalanobis donnée. Par conséquent,

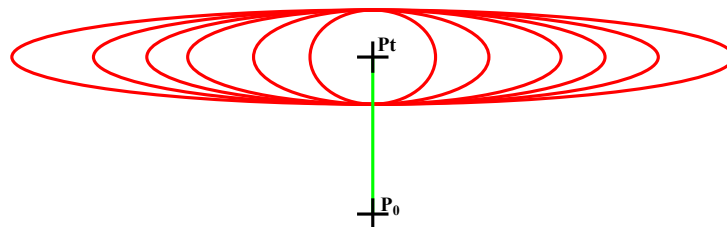


FIGURE 3.25 : Tous les ellipsoïdes, en vue de dessus, possèdent la même distance de Mahalanobis vis-à-vis du point P_0 .

nous avons décidé de traiter chaque dimension de manière indépendante afin qu'elles ne se compensent pas les unes avec les autres : il s'agit alors d'une distance de Mahalanobis sur chaque dimension ce qui revient à seuiller les variances.

En prenant comme seuil la valeur pour un point à une distance donnée par rapport à la position de première observation, on peut dresser le tableau de valeurs 3.3 valables pour chaque dimension. De cette manière, les points loin sont conservés. Le

	Référence	Distance d'application					
Distance	1 m	5 m	10 m	50 m	100 m	500 m	1 km
Seuil	10 cm	50 cm	1 m	5 m	10 m	50 m	100 m

TABLE 3.3 : Tableaux des incertitudes de seuil pour une référence de 10 cm^3 à 1 m .

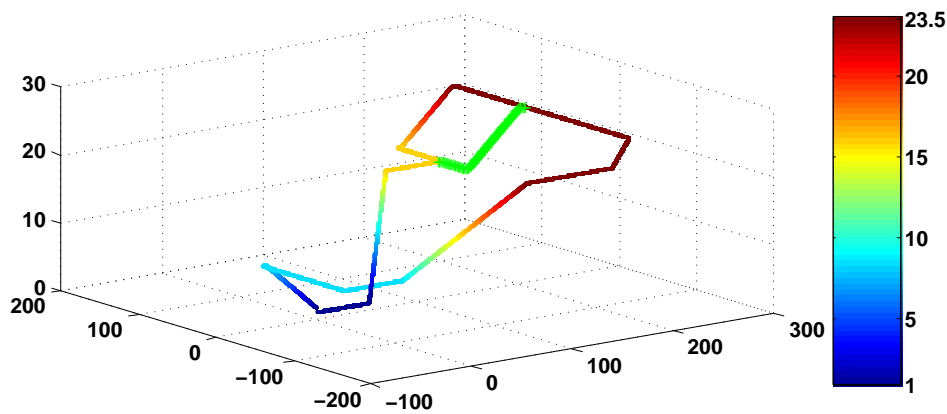
temps de survie d'un point suite à une succession de non-observations doit à présent être défini. Une méthode cohérente serait de le faire sous forme bayésienne, en calculant une probabilité de non-observation ou d'occultation. Ce travail reste ouvert et c'est une méthode de comptabilisation d'erreur par incrémentation/décrémentation qui a été implantée. Cette méthode, similaire à celle employée dans le protocole CAN (Controller Area Network), se traduit par une valeur d'incrément différent de celle de décrémentation. Ainsi, c'est la succession d'erreurs, ou dans notre cas de non-observations, qui entraîne le retrait du point du vecteur d'état. Ce retrait est assujéti au critère de conservation décrit précédemment. Le protocole mis en place permet de perdre momentanément un point tant que les observations suivantes sont suffisantes pour venir annuler l'incrément. En posant les limites, on trouve les valeurs d'incrément et de décrémentation de la manière suivante :

- la décrémentation est unitaire ;
- si un point est observé dans n images successives alors son indice d'erreur doit être décrémentation d'un facteur n ;
- si un point n'est pas observé pendant m images successives, alors il doit être retiré.

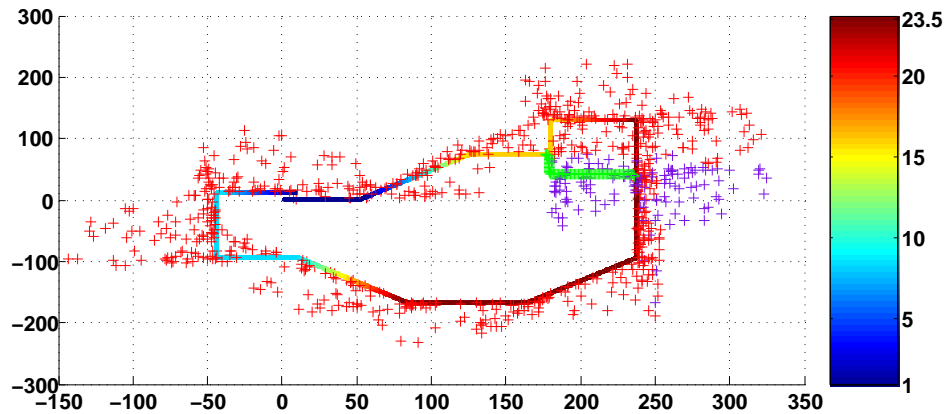
On en déduit alors aisément que l'incrémentaion sera de n , la décrémentaion sera de 1 et que le seuil de retrait est de $m \times n$.

3.8 Résultats de simulation

Pour réaliser un environnement de simulation cohérent avec un échappement de trajectoire, on se propose de modifier le scénario existant. La figure 3.26 montre un tronçon de déviation qui a été introduit dans la trajectoire. De plus, un environnement supplémentaire a été généré pour fournir des points à localiser. Le résultat



(a) La trajectoire de référence et la portion de contournement (portion verte).



(b) Les trajectoires et leurs environnements respectifs.

FIGURE 3.26 : La trajectoire de contournement est symbolisée par la portion verte et l'environnement supplémentaire qui alimentera le SLAM est représenté par les points violets.

de l'exécution de la fonction de localisation sans introduction de la partie SLAM est présenté dans la figure 3.27. On remarque alors que, lors de la trajectoire de contournement, n'ayant plus de points de la carte de référence, le système est incapable d'estimer correctement son altitude et son orientation (principalement son

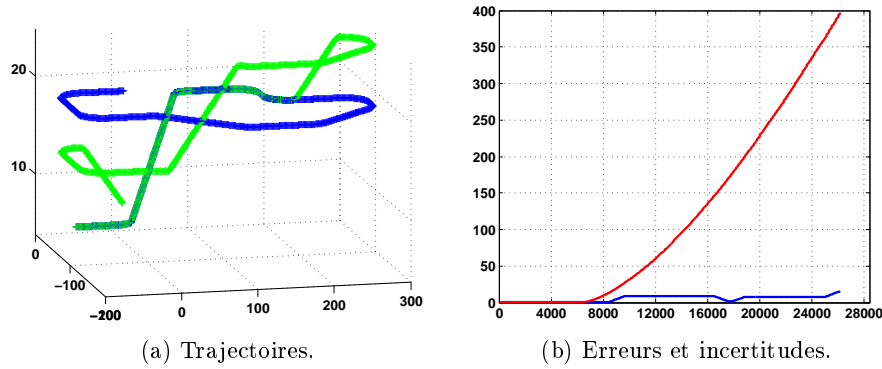


FIGURE 3.27 : Modification de la trajectoire de référence sans introduction du SLAM. Sur la figure 3.27a sont représentées la trajectoire réelle en vert et la trajectoire reconstruite en bleu. La figure 3.27b donne la distance euclidienne entre les deux trajectoires (en bleu) et l'incertitude à 1σ (en rouge) pour chaque pose du véhicule (axe des abscisses).

tangage). Dès lors, l'estimation de la pose du robot est réalisée uniquement à partir des données proprioceptives. Il s'en suit la perte complète de la localisation du véhicule au sein de la carte de référence ainsi qu'une erreur en localisation de l'ordre d'une dizaine de mètres pour une incertitude autant intègre que gigantesque.

A présent, voyons le comportement du système lorsqu'on mélange la localisation dans la carte de référence (comme précédemment) et le processus SLAM. La réalisation de la déviation de la trajectoire est représentée dans la figure 3.28 tandis que la création de carte par SLAM pur comme le ferait le premier véhicule a été décrite par la figure 3.24b. On remarque alors sur la figure 3.28b deux augmentations notables

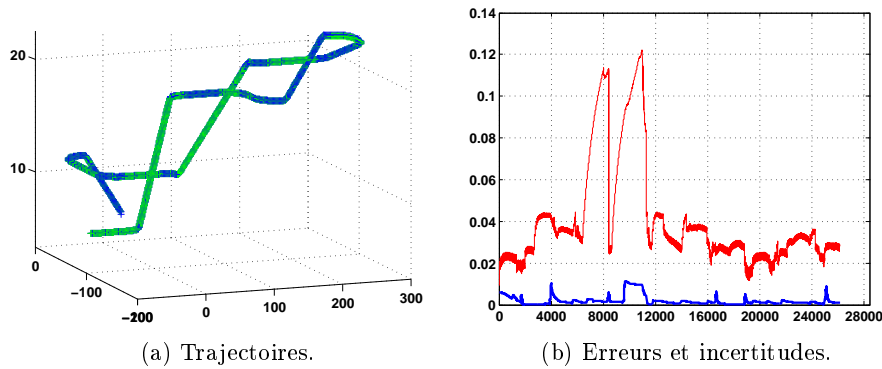


FIGURE 3.28 : Modification de la trajectoire de référence avec introduction du SLAM. Sur la figure 3.28a sont représentées la trajectoire réelle en vert et la trajectoire reconstruite en bleu. La figure 3.28b donne la distance euclidienne entre les deux trajectoires (en bleu) et l'incertitude à 1σ (en rouge).

de l'incertitude qui sont dues aux transitions entre les approches. Le premier se situe au moment où l'algorithme commence à générer sa propre carte pour poursuivre la localisation. Dès lors, il n'y a plus suffisamment de points de la carte de référence et les points de la carte SLAM n'ont pas encore convergé. Cette première augmentation se réduit et laisse place à une suivante qui traduit l'introduction du biais. On constate l'efficacité de la méthode tandis que le véhicule réussit à se localiser avec la carte de référence après avoir suivi une trajectoire de contournement n'appartenant pas à la trajectoire de référence.

3.9 Conclusion

En conclusion de ce chapitre, nous rappellerons les différentes optimisations et corrections mises en place afin de pouvoir faire cohabiter deux cartes de nature intrinsèquement différente. Le principe était de permettre à un véhicule de pouvoir se localiser tandis qu'il s'éloigne temporairement de la trajectoire de référence. Cela impliquait de créer une carte supplémentaire (carte SLAM), et de pouvoir maintenir une relation de cohérence avec la carte de référence.

Cette cohabitation se traduit par la prise en compte d'un biais qui correspond au lien entre les deux cartes. Son estimation se fait en intégrant ses paramètres au sein du vecteur d'état. Son évolution au cours des différentes observations permet de pouvoir s'écarter de la trajectoire initiale pour ensuite venir la reprendre un peu plus loin.

Cette technique de SLAM pourrait, sans la prise en compte du biais, être celle utilisée par le véhicule dont la charge est de définir la trajectoire et de construire la carte de référence. La prise en compte de l'expression de la non-linéarité offre une réponse satisfaisante au problème du filtre de Kalman où les fonctions d'observations sont sous la forme d'un quotient de deux formes rationnelles du premier ordre.

D'un point de vue pratique, cette correction permet d'augmenter la durée de vie d'un point dans le vecteur d'état et de diminuer, par conséquent, le nombre d'initialisation de points coûteuse en temps de calcul. Des résultats sur données réelles seront présentés dans le chapitre 5.

Il est à noter que la vitesse de convergence des points dépend principalement de deux facteurs : le nombre d'observations ainsi que le déplacement exécuté par le véhicule au cours de la durée d'existence du point dans le vecteur d'état. Ajouter une autre caméra sur le véhicule permet d'avoir simultanément deux observations du même point, à condition de pouvoir le reconnaître correctement et d'avoir une paramétrisation de la position de la deuxième caméra par rapport à la première. La stéréoscopie est fortement contrainte par la calibration des deux caméras ainsi que par la distance qui les sépare. Par conséquent, la solution bi-monoculaire sera privilégiée. Cette étude ainsi que l'intégration dans un système à deux caméras des différents correctifs développés fait l'objet du chapitre 4.

SLAM bi-monoculaire

Sommaire

4.1	Introduction et rappels	106
4.2	Relations d'observation et d'évolution	107
4.2.1	Relations d'observation	107
4.2.2	Relations d'évolution	109
4.3	Résultats	110
4.3.1	Résultats de localisation d'un point	110
4.3.2	Résultats d'autocalibration	112
4.4	Conclusion	114

4.1 Introduction et rappels

Au montage monoculaire précédemment décrit, on ajoute une deuxième caméra dont le champ de vision recouvre partiellement celui de la première. Le principe du bi-monoculaire consiste à considérer cette deuxième caméra comme un autre véhicule dont les paramètres de position et d'orientation sont fortement liés à la première. Les deux caméras étant montées sur le même véhicule, il s'agit de donner l'équation des paramètres extrinsèques la seconde caméra dans un repère judicieux.

Une première approche serait de la considérer dans le repère véhicule mais cela ne permet pas de mettre en place le lien statistique qu'il y a entre les deux caméras. De plus, cela impose un changement de repère introduisant l'erreur sur les paramètres extrinsèques de la première caméra. L'approche de la géométrie épipolaire offre une représentation élégante du problème. En effet, dans le problème d'observation d'un même point par plusieurs caméras, le repère est défini comme celui de la première caméra. Cette représentation a l'avantage de ne pas être tributaire de l'erreur de position et d'orientation de la caméra, origine du repère, dans le repère véhicule. Ainsi, les coordonnées de la seconde caméra seront définies par rapport à la première ce qui va permettre de les lier statistiquement.

Dans ce chapitre, nous allons montrer principalement deux choses : comment l'intégration d'une seconde caméra augmente la vitesse de convergence et comment il est possible de trouver la position relative des deux caméras à une incertitude près. Ce deuxième point s'avère particulièrement intéressant pour les calibrations de banc stéréoscopique. En effet, le calcul des paramètres intrinsèques de chaque caméra peut être fait sans que les capteurs ne soient installés sur le véhicule. Cependant, cette implantation est nécessaire pour la calibration des paramètres extrinsèques. Par ailleurs, l'interprétation de l'information en stéréoscopie est fortement liée à cette calibration délicate. La méthode, déjà explorée par [Solà 2007], permet d'obtenir rapidement, et avec une incertitude connue les paramètres d'un banc stéréoscopique.

De manière générale, on s'attend à une localisation plus rapide des points avec l'approche bi-monoculaire. Cette particularité permettra d'obtenir des cartes précises plus rapidement et réduira du coup la dérive du facteur d'échelle, jusque là uniquement maintenu par l'odométrie. Par ailleurs, on s'attend aussi à ce que le système fasse converger les paramètres extrinsèques de la seconde caméra vers leur valeur réelle, déterminant du même coup la pose de cette caméra dans le repère robot.

Après avoir rappelé les équations d'observation de la première caméra, déjà présentées précédemment, nous verrons de quelle manière introduire les paramètres de la seconde caméra pour obtenir une nouvelle fonction d'observation. A partir d'expérimentations réelles, nous présenterons des résultats d'observation à une caméra puis à deux. Dans chacune des méthodes, notre approche du calcul des zones de recherche ainsi que la correction du gain de Kalman seront implémentées, leurs améliorations respectives ayant été étudiées précédemment.

4.2 Relations d'observation et d'évolution

4.2.1 Relations d'observation

Deux caméras sont placées sur un même véhicule. Les paramètres intrinsèques de chaque caméra sont connus par calibration et fournissent les matrices \mathbf{K} et \mathbf{K}' , respectivement pour les caméras C_1 et C_2 . Les paramètres extrinsèques de la caméra C_1 sont définis dans le repère véhicule tandis que ceux de la caméra C_2 sont définis dans le repère de C_1 . Ces définitions sont illustrées par la figure 4.1.

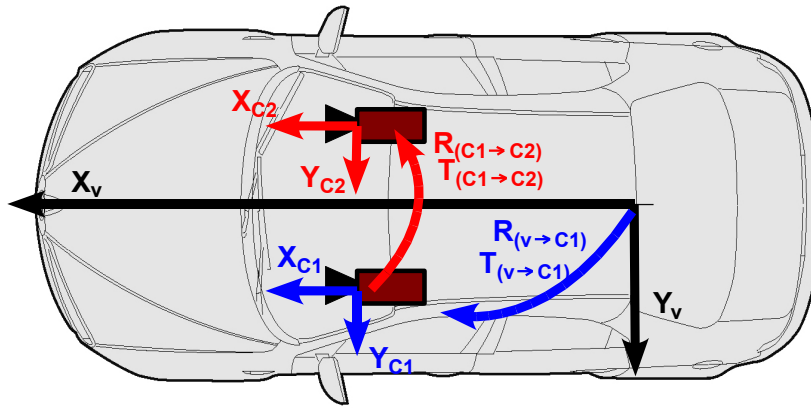


FIGURE 4.1 : Les paramètres de la pose de la caméra 1 (en bleu) sont définis dans le repère véhicule. Les paramètres de la pose de la caméra 2 (en rouge) sont définis dans le repère de la caméra 1.

Rappelons dans un premier temps l'équation du modèle d'observation dans la première caméra C_1 :

$$u = \frac{\vec{K}_2 \mathbf{R}_{(v \rightarrow C_1)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow C_1)})}{\vec{K}_1 \mathbf{R}_{(v \rightarrow C_1)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow C_1)})} = \frac{Nu}{De} \quad (4.1)$$

$$v = \frac{\vec{K}_3 \mathbf{R}_{(v \rightarrow C_1)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow C_1)})}{\vec{K}_1 \mathbf{R}_{(v \rightarrow C_1)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow C_1)})} = \frac{Nv}{De} \quad (4.2)$$

Avec la représentation choisie, il suffit d'introduire, avant la projection par la matrice de paramètres intrinsèques, le passage du repère de la caméra 1 à la caméra 2. Soit $\vec{C}_2 = (c_x, c_y, c_z, c_{\alpha_x}, c_{\alpha_y}, c_{\alpha_z})^T$ le vecteur de paramètres extrinsèques qui définissent la caméra 2 dans le repère \mathcal{R}_{C_1} de la première. Etant donné que ces coefficients vont être précisés, ils seront intégrés au vecteur d'état. Avec $\mathbf{R}_{(C_1 \rightarrow C_2)}$ la matrice de rotation et $\vec{T}_{(C_1 \rightarrow C_2)}$ le vecteur de translation définissant C_2 dans C_1 ,

on trouve les équations d'observation suivantes :

$$u' = \frac{\vec{K}'_2 \mathbf{R}_{(C_1 \rightarrow C_2)}^T \left(\mathbf{R}_{(v \rightarrow C_1)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow C_1)}) - \vec{T}_{(C_1 \rightarrow C_2)} \right)}{\vec{K}'_1 \mathbf{R}_{(C_1 \rightarrow C_2)}^T \left(\mathbf{R}_{(v \rightarrow C_1)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow C_1)}) - \vec{T}_{(C_1 \rightarrow C_2)} \right)} = \frac{Nu'}{De'} \quad (4.3)$$

$$v' = \frac{\vec{K}'_3 \mathbf{R}_{(C_1 \rightarrow C_2)}^T \left(\mathbf{R}_{(v \rightarrow C_1)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow C_1)}) - \vec{T}_{(C_1 \rightarrow C_2)} \right)}{\vec{K}'_1 \mathbf{R}_{(C_1 \rightarrow C_2)}^T \left(\mathbf{R}_{(v \rightarrow C_1)}^T (\mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t - \vec{T}_{(m \rightarrow v)}) - \vec{T}_{(v \rightarrow C_1)}) - \vec{T}_{(C_1 \rightarrow C_2)} \right)} = \frac{Nv'}{De'} \quad (4.4)$$

Avec le formalisme matriciel, il faut calculer les jacobienues par rapport aux paramètres du véhicule, à ceux du point, et à présent aux paramètres extrinsèques de la deuxième caméra. Puisque les coefficients de \vec{C}_2 n'interviennent pas dans les relations 4.1 et 4.2, les jacobienues sont donc celles développées précédemment dans la section 2.3.2.4. Pour la relation 4.3, les jacobienues sont les suivantes :

$$\begin{cases} \frac{\partial u'}{\partial X_v} = -\frac{\partial u'}{\partial x_n} = C_u \mathbf{R}_{(c_1 \rightarrow c_2)}^T \mathbf{R}_{(v \rightarrow c_1)}^T \mathbf{R}_{(m \rightarrow v)}^T (-1 \ 0 \ 0)^T \\ \frac{\partial u'}{\partial Y_v} = -\frac{\partial u'}{\partial y_n} = C_u \mathbf{R}_{(c_1 \rightarrow c_2)}^T \mathbf{R}_{(v \rightarrow c_1)}^T \mathbf{R}_{(m \rightarrow v)}^T (0 \ -1 \ 0)^T \\ \frac{\partial u'}{\partial Z_v} = -\frac{\partial u'}{\partial z_n} = C_u \mathbf{R}_{(c_1 \rightarrow c_2)}^T \mathbf{R}_{(v \rightarrow c_1)}^T \mathbf{R}_{(m \rightarrow v)}^T (0 \ 0 \ -1)^T \\ \frac{\partial u'}{\partial \alpha_x} = C_u \mathbf{R}_{(c_1 \rightarrow c_2)}^T \mathbf{R}_{(v \rightarrow c_1)}^T \left(\frac{\partial \mathbf{R}_{(m \rightarrow v)}}{\partial \alpha_x} \right)^T (\vec{P}t_n - \vec{T}_{(m \rightarrow v)}) \\ \frac{\partial u'}{\partial \alpha_y} = C_u \mathbf{R}_{(c_1 \rightarrow c_2)}^T \mathbf{R}_{(v \rightarrow c_1)}^T \left(\frac{\partial \mathbf{R}_{(m \rightarrow v)}}{\partial \alpha_y} \right)^T (\vec{P}t_n - \vec{T}_{(m \rightarrow v)}) \\ \frac{\partial u'}{\partial \alpha_z} = C_u \mathbf{R}_{(c_1 \rightarrow c_2)}^T \mathbf{R}_{(v \rightarrow c_1)}^T \left(\frac{\partial \mathbf{R}_{(m \rightarrow v)}}{\partial \alpha_z} \right)^T (\vec{P}t_n - \vec{T}_{(m \rightarrow v)}) \\ \frac{\partial u'}{\partial c_x} = C_u \mathbf{R}_{(c_1 \rightarrow c_2)}^T (-1 \ 0 \ 0)^T \\ \frac{\partial u'}{\partial c_y} = C_u \mathbf{R}_{(c_1 \rightarrow c_2)}^T (0 \ -1 \ 0)^T \\ \frac{\partial u'}{\partial c_z} = C_u \mathbf{R}_{(c_1 \rightarrow c_2)}^T (0 \ 0 \ -1)^T \\ \frac{\partial u'}{\partial c_{\alpha_x}} = C_u \left(\frac{\partial \mathbf{R}_{(c_1 \rightarrow c_2)}}{\partial c_{\alpha_x}} \right)^T \mathbf{R}_{(v \rightarrow c_1)}^T \mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t_n - \vec{T}'_{(m \rightarrow v)}) \\ \frac{\partial u'}{\partial c_{\alpha_y}} = C_u \left(\frac{\partial \mathbf{R}_{(c_1 \rightarrow c_2)}}{\partial c_{\alpha_y}} \right)^T \mathbf{R}_{(v \rightarrow c_1)}^T \mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t_n - \vec{T}'_{(m \rightarrow v)}) \\ \frac{\partial u'}{\partial c_{\alpha_z}} = C_u \left(\frac{\partial \mathbf{R}_{(c_1 \rightarrow c_2)}}{\partial c_{\alpha_z}} \right)^T \mathbf{R}_{(v \rightarrow c_1)}^T \mathbf{R}_{(m \rightarrow v)}^T (\vec{P}t_n - \vec{T}'_{(m \rightarrow v)}) \end{cases} \quad (4.5)$$

avec $C_u = \frac{De' \vec{K}'_2 - Nu' \vec{K}'_1}{De'^2}$. Les jacobienues de la relation 4.4 sont obtenues en remplaçant C_u par $C_v = \frac{De' \vec{K}'_3 - Nv' \vec{K}'_1}{De'^2}$. Comme précédemment, ces jacobienues vont permettre de mettre à jour les différents éléments du vecteur d'état au travers du filtre de Kalman étendu en fonction de chaque caméra. Une première expérimentation peut être menée avec la trajectoire de simulation. En plaçant deux caméras dont les champs se recouvrent sur le véhicule 4.2, on réalise alors un système d'observation bi-monoculaire.

Les équations d'observation ayant été explicitées pour chacune des caméras, nous allons présenter un scénario et expliquer l'amélioration d'une telle configuration. Par



FIGURE 4.2 : Position des caméras (cercles rouges) et du GPS (cercle vert).

ailleurs, étant donné que l'on va chercher à améliorer les paramètres extrinsèques de la caméra C_2 , ceux-ci doivent être intégrés au vecteur d'état. Enfin, terminons sur l'algorithme 4 qui présente la méthode employée.

Algorithme 4 Algorithme bi-monoculaire

Précondition : une image de chaque caméra et les données proprioceptives relatives

- 1 : Prédiction de Kalman
 - 2 : Recherche d'informations relatives à la caméra 1
 - 3 : Mise à jour de la position du véhicule et de chaque point observé et présent dans le vecteur d'état
 - 4 : Recherche d'informations relatives à la caméra 2
 - 5 : Mise à jour de la position du véhicule, de la position de la caméra 2 et de chaque point observé et présent dans le vecteur d'état
 - 6 : **Retour** (la localisation du véhicule, de la caméra 2 ainsi que la carte des points ayant convergé)
-

4.2.2 Relations d'évolution

Comme nous l'avons vu précédemment, le vecteur de paramètres extrinsèques de la caméra 2 va être ajouté dans le vecteur d'état. Par conséquent, il faut définir un modèle d'évolution. Les caméras sont rigidement liées au véhicule et les variations de poses de ces dernières ne peuvent être estimées à partir des données proprioceptives. La fonction d'évolution est donc une fonction identité :

$$f_c \left(c_{x_{k+1|k}}, c_{y_{k+1|k}}, c_{z_{k+1|k}}, c_{\alpha_{x_{k+1|k}}}, c_{\alpha_{y_{k+1|k}}}, c_{\alpha_{z_{k+1|k}}} \right) = \dots$$

$$\left(c_{x_{k|k}}, c_{y_{k|k}}, c_{z_{k|k}}, c_{\alpha_{x_{k|k}}}, c_{\alpha_{y_{k|k}}}, c_{\alpha_{z_{k|k}}} \right) \quad (4.6)$$

où f_c désigne la fonction d'évolution des paramètres extrinsèques de la caméra 2. Par la suite, on cherchera à calibrer le système. Par conséquent, il n'y a pas de bruit d'évolution à définir sur ces paramètres. Cependant, dans une optique d'auto-calibration pour de longues missions, on peut définir un bruit d'évolution qui sera

blanc, gaussien, d'espérance nulle et d'écart-type inférieur à la tolérance voulue. Cette méthode réduit aussi les effets de sur-convergence propre au filtre de Kalman.

La configuration bi-monoculaire sera déployée de deux façons distinctes. Premièrement, de manière à constater l'avantage du bi-monoculaire sur le monoculaire, nous traiterons le cas de la localisation. Dans un deuxième temps, la calibration des paramètres extrinsèques de la seconde caméra sera abordée. Pour chacune des deux approches, des résultats seront montrés dans la section suivante.

4.3 Résultats

Deux caméras sont installées sur un Cycab et déployées en configuration bi-monoculaire (expression de la seconde caméra dans le repère de la première et les champs d'observation des caméras se recouvrent). Sur une trajectoire de l'environnement PAVIN, on fait évoluer le véhicule tandis qu'il observe son monde avec soit la caméra 1 seule, soit les deux caméras simultanément. La synchronisation des caméras étant assurée par un module indépendant et les cadences des capteurs proprioceptifs étant plus importantes que la fréquence de chaque caméra, on peut considérer avoir, à chaque couple d'images, les données odométriques et d'angle de roues correspondantes. Pour montrer l'intérêt d'une telle méthode, nous allons déployer le système en tentant de cartographier un point dont la position a été préalablement définie dans le repère de la première observation du véhicule (soit la position définissant l'origine du repère monde).

4.3.1 Résultats de localisation d'un point

Sur une trajectoire en ligne droite d'environ 6 mètres, on exécute le processus SLAM uniquement avec la caméra 1. L'algorithme localise le véhicule et cartographie simultanément un point (rappelons que la position de ce point dans le repère monde est connue). La première image perçue est donnée par la figure 4.3.

Le résultat de cette cartographie est donné par la figure 4.4. Après 60 itérations, le point est toujours observable (une analyse sur données réelles en utilisant ce même point sera dévoilée dans le chapitre suivant, justifiant l'intérêt de la méthode de calcul des zones de recherche).

On remarque que le point converge effectivement vers la bonne valeur tandis que le véhicule se localise. A présent, on ajoute l'observation via la seconde caméra et du même coup, le vecteur d'état contiendra, en plus des paramètres de position et d'orientation du véhicule, les paramètres extrinsèques de la caméra 2 (exprimée dans le repère de la caméra 1). On s'attend ici à faire converger la position de la caméra 2, la position du point cartographié ainsi que la localisation du véhicule.

A la première itération, nous avons donc deux observations du monde comme le montre la figure 4.5.

De la même manière que précédemment, la figure 4.6 montre la convergence de la position du point, sur le rayon de vue, vers le point réel cherché.

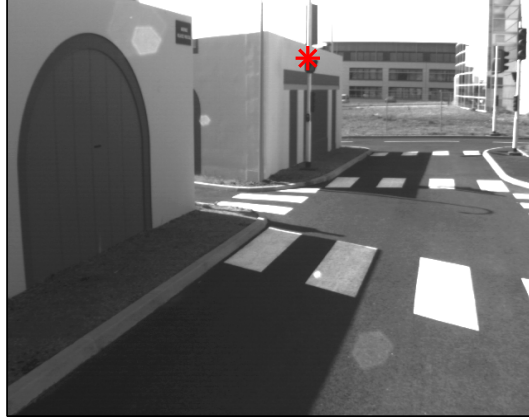


FIGURE 4.3 : Image de la première observation avec la position dans l'image du point à localiser (croix rouge).

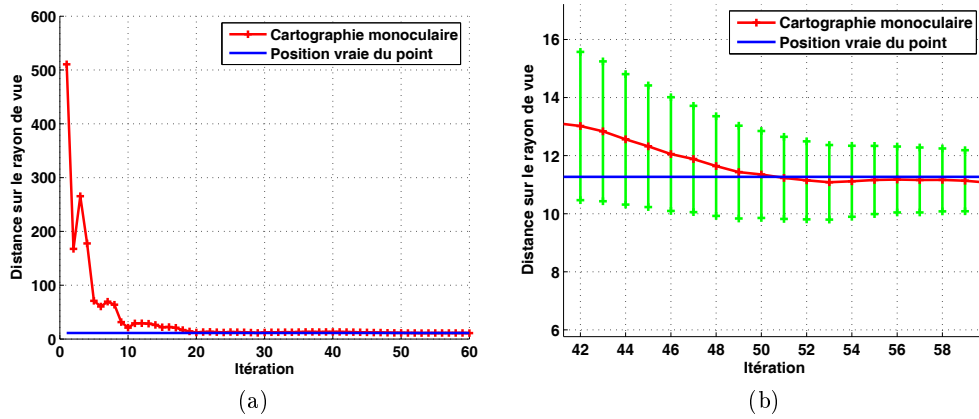


FIGURE 4.4 : La figure illustre la convergence du point fictif vers le point réel. En rouge, la position du point estimée sur le rayon de vue et en bleu, la valeur à atteindre (11.27 mètres). Sur la figure de droite, un zoom de la figure de gauche pour les dernières itérations (l'écart type à 3σ de chaque estimation est représenté en vert). L'écart à l'itération 60 n'est plus que de 13 cm entre la valeur vraie et la valeur estimée.

Pour plus de clarté dans la comparaison, la figure 4.7 montre sur un même graphique la courbe de convergence avec chacune des deux approches.

On constate alors que l'approche bi-monoculaire converge plus rapidement. Par ailleurs, la figure 4.8 donne, sur le même principe, la courbe de convergence des incertitudes de chaque positionnement. On peut nettement constater que l'incertitude de l'approche bi-monoculaire est toujours inférieure à l'incertitude de l'approche monoculaire.

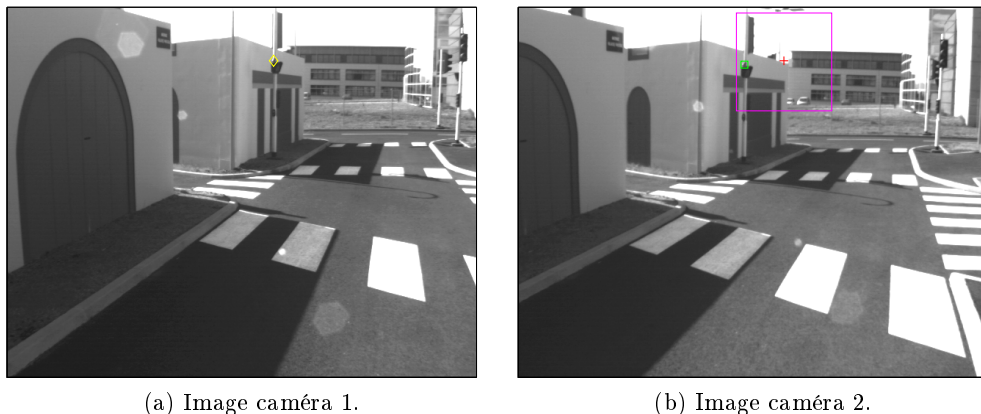


FIGURE 4.5 : A gauche, l'image de la première observation de la caméra 1 avec la position, dans l'image, du point à localiser (losange jaune). A droite, on retrouve la projection du point 3D (croix rouge), la zone de recherche de l'observation (cadre magenta) et le résultat de l'association de données (carré vert).

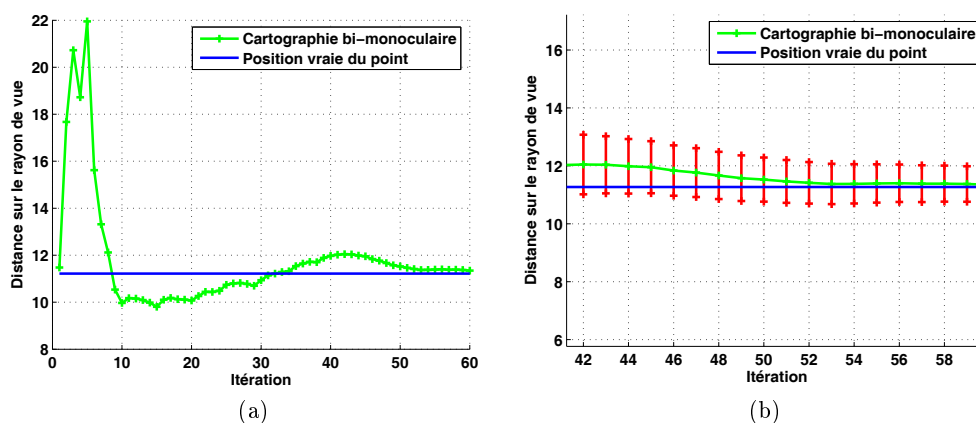


FIGURE 4.6 : La figure illustre la convergence du point fictif vers le point réel. En vert, la position du point estimée sur le rayon de vue et en bleu, la valeur à atteindre (11.27 mètres). Sur la figure de droite, un zoom de la figure de gauche pour les dernières itérations (l'écart type à 3σ de chaque estimation est représenté en rouge). L'écart à l'itération 60 n'est plus que de 10 cm entre la valeur vraie et la valeur estimée.

4.3.2 Résultats d'autocalibration

Dans cette approche, le filtre de Kalman permet de calculer la position estimée d'une caméra et de fournir une incertitude à cette localisation. La position réelle de la caméra étant parfaitement établie, nous pouvons aussi vérifier la convergence de ses paramètres extrinsèques vers les valeurs cherchées (voir figure 4.10).

De plus, la pose de la deuxième caméra étant relativement précise dans le repère

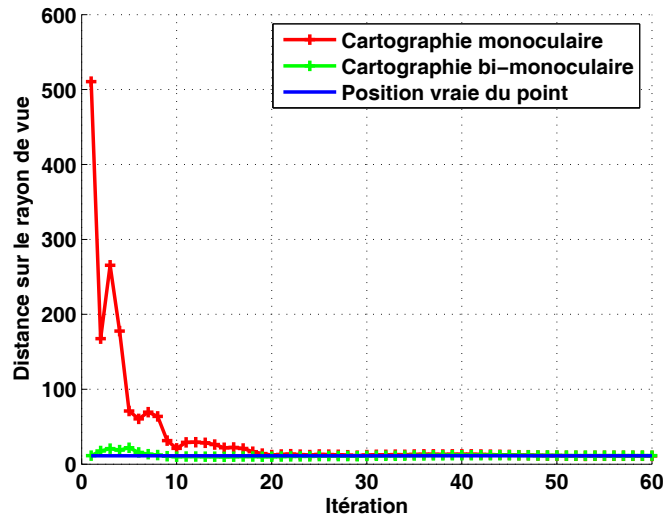


FIGURE 4.7 : Courbes de convergence des deux approches.

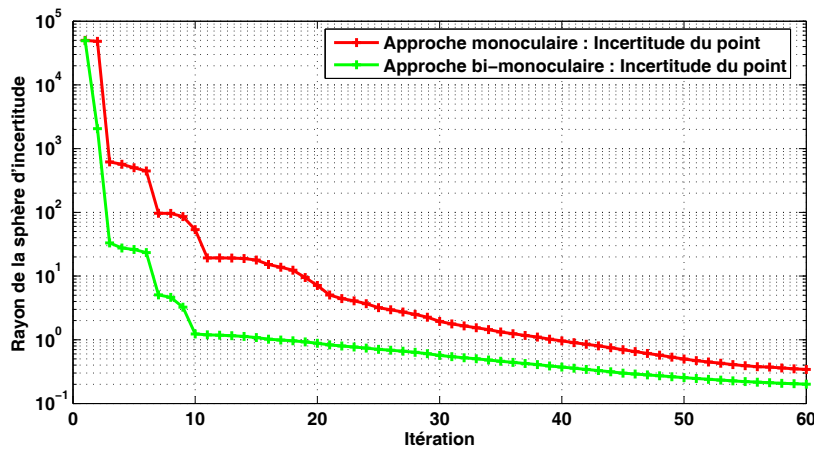


FIGURE 4.8 : La figure montre l'évolution de l'incertitude du point suivant le rayon de vue de chaque approche.

véhicule, le facteur d'échelle est directement lié à la distance qui sépare les deux caméras. Dans l'approche monoculaire, ce facteur d'échelle était lié à l'odométrie. Par conséquent, on pourrait envisager un capteur odométrique fortement bruité pour voir le comportement de chaque algorithme. On s'attend ici à voir une nette amélioration de la version bi-monoculaire par rapport à la version monoculaire.

La réalisation d'un processus monoculaire et bi-monoculaire avec une odométrie fortement bruitée a été menée. La figure 4.11 montre la convergence de la localisation d'un point par un processus monoculaire dont l'odométrie a été fortement bruitée (à hauteur de 50 cm/m).

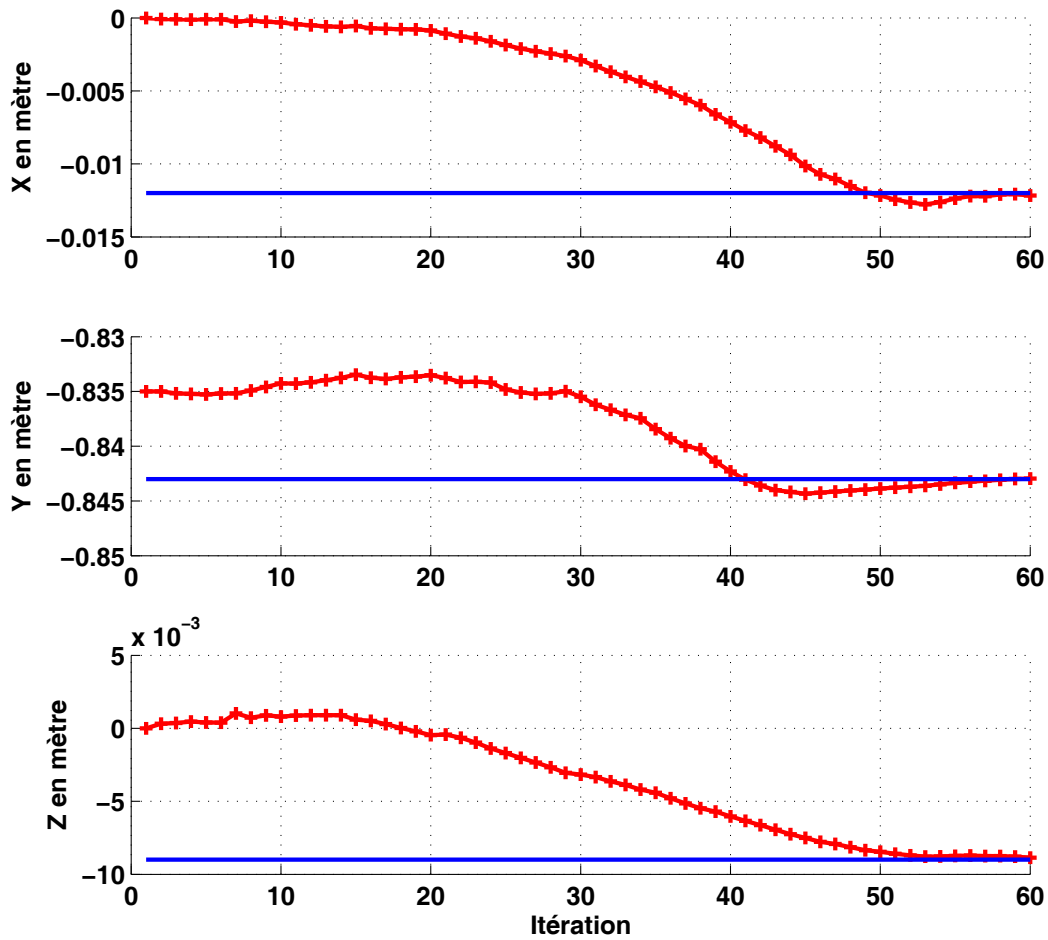


FIGURE 4.9 : La figure montre la convergence des paramètres extrinsèques de translation vers la valeur cherchée (l'incertitude a été initialisée à 2 cm).

4.4 Conclusion

Finalement, nous avons développé une approche bi-monoculaire pour simultanément cartographier et se localiser. L'approche exécutée ici sur un point reste valable lorsqu'on en localise un nombre plus important. Dès lors, les paramètres de la caméra 2 convergent plus rapidement et les mises à jour deviennent de plus en plus négligeables si bien qu'au-delà de 5 itérations, ils ne sont quasiment plus modifiés (le processus se comporte comme si les paramètres extrinsèques de la seconde caméra étaient parfaitement connus). Il convient cependant, afin de ne pas faire sur-converger le processus, de ne pas appliquer des corrections inférieures au millimètre. En effet, puisqu'il existe toujours une faible incertitude, le système est susceptible d'évoluer jusqu'à perdre l'intégrité.

Par ailleurs, on constate que l'approche monoculaire converge vers la bonne solution assez rapidement et les différentes corrections appliquées permettent :

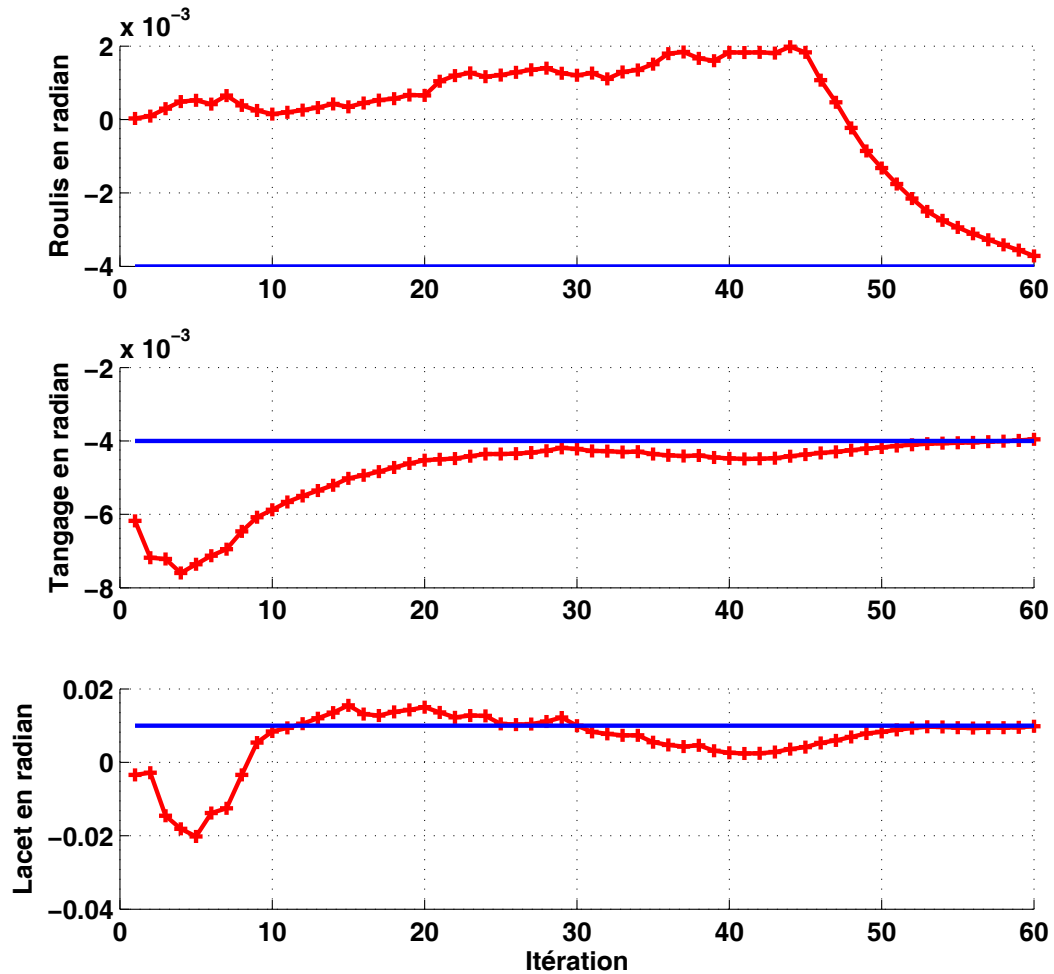


FIGURE 4.10 : La figure montre la convergence des paramètres extrinsèques de rotation vers la valeur cherchée (l'incertitude a été initialisée à 2 degrés).

- une mise à jour plus cohérente ;
- une détection de l'observation sur un plus grand nombre d'images ;
- une meilleure intégrité.

Les améliorations apportées à l'approche monoculaire permettent d'obtenir des résultats compétitifs vis-à-vis de l'approche bi-monoculaire. Cependant, cette dernière permet une bien meilleure estimation de la localisation lorsque l'odométrie est fortement bruitée ou lorsque le nombre d'observation d'un point diminue. Ce cas intervient avec l'augmentation de la vitesse du véhicule. Par ailleurs, l'approche bi-monoculaire permet une autocalibration de la paire de caméra et assiste l'odométrie dans le maintien du facteur d'échelle.

En conclusion, l'approche bi-monoculaire est pertinente dès lors que seul un faible nombre d'images est disponible pour localiser un point comme cela peut être le cas à des vitesses plus importantes. Cependant, la convergence du monoculaire,

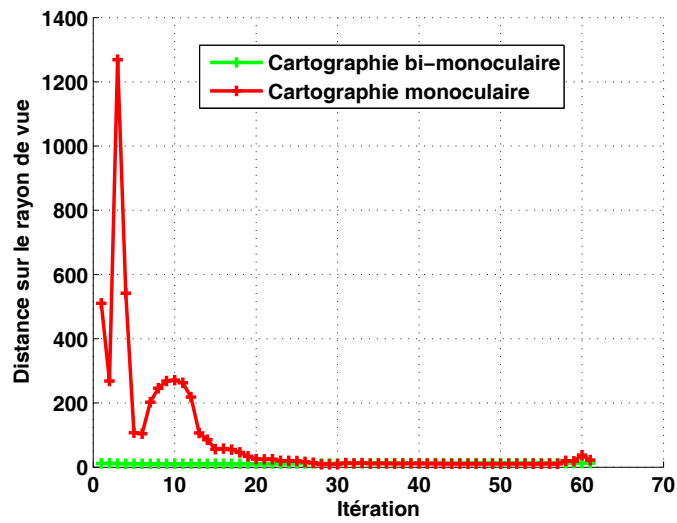


FIGURE 4.11 : Courbes de convergence pour une odométrie fortement bruitée. On remarque que l'approche monoculaire met beaucoup plus d'itérations pour converger contrairement l'approche bi-monoculaire. Il faut attendre l'itération 30 pour que les deux courbes soient séparées de moins d'un mètre.

dont le facteur d'échelle est maintenu par l'odométrie, reste tout aussi fiable et sera conservée pour les essais sur données réelles avec des trajectoires plus importantes.

Expérimentations

Sommaire

5.1	SLAM sur le site PAVIN	118
5.1.1	Définition des zones de recherche	119
5.1.2	A10 : SLAM sans aucune amélioration	121
5.1.3	A11 : SLAM avec l'initialisation « monde plan »	121
5.1.4	A12 : SLAM avec l'initialisation « monde plan » et la correction des zones de recherche	121
5.1.5	A13 : SLAM avec l'initialisation « monde plan », correction des zones de recherche et correction du gain de Kalman	122
5.1.6	Comparatif des algorithmes	123
5.2	Localisation à partir d'une carte de référence sur le site PAVIN	124
5.3	SLAM avec carte de référence	124
5.4	Conclusion générale des expérimentations	127

Les chapitres précédents ont présenté les différentes problématiques de la non-linéarité de la fonction d'observation introduite par le modèle pinhole d'une caméra. Nous avons pu déceler ses effets dans le calcul de la zone de recherche dans l'image, dans l'initialisation d'un point ainsi que dans le calcul du gain de Kalman.

Nous avons abordé plusieurs problématiques liées notamment à la non-linéarité de la fonction d'observation. Nous avons vu une nouvelle méthode de calcul des zones de recherche dans l'image ainsi qu'une méthode de correction du gain de Kalman. De plus, une approche bi-monoculaire a été développée pour pallier un manque d'observation qui peut être induit, par exemple, par une augmentation de la vitesse du véhicule. Par ailleurs, cette approche permet aussi de trouver les paramètres extrinsèques d'un banc stéréoscopique et de les maintenir à jour au cours d'une mission. Nous allons à présent expérimenter les algorithmes de localisation et de SLAM sur des données réelles en incluant au fur et à mesure, les diverses améliorations. Dans un premier temps, nous allons élaborer une carte de référence à partir d'un algorithme SLAM, puis nous montrerons que nous pouvons naviguer dans celle-ci avec l'algorithme de localisation en respectant les contraintes. Enfin, nous traiterons le scénario de déviation de la trajectoire de référence.

5.1 SLAM sur le site PAVIN

Les travaux du chapitre 3 ont permis de développer une approche permettant de se localiser dans une carte de référence, et de tenir à jour simultanément une seconde carte via un processus SLAM. Les équations prenant en compte la dérive de localisation ont été développées et permettent le maintien de la cohérence entre les deux cartes.

Si l'on suggère le cas où la carte de référence est vide, on se trouve tout simplement dans une approche SLAM qui pourrait être le principe de construction de la carte de référence. Dans le cadre d'un convoi par exemple, un premier véhicule explore un chemin et définit sa trajectoire par rapport à la carte SLAM qu'il construit. Cette carte SLAM sera alors la carte de référence pour le ou les véhicules qui souhaiteront suivre la même route : les véhicules suiveurs n'auront qu'à exécuter l'algorithme de localisation développé dans le chapitre 2. Nous allons mettre ce scénario en place dans l'environnement PAVIN, en ajoutant au fur et à mesure les améliorations présentées.

Dans un premier temps, il faut élaborer la carte et la trajectoire. La figure 5.1 donne le tracé suivi par un opérateur à bord d'un véhicule sur le site PAVIN. Cette trajectoire mesure 170 mètres et sera réalisée avec une vitesse maximale de 2 mètres par seconde.

Nous allons effectuer sur ce tracé 4 algorithmes de SLAM différents présentant à chaque fois une évolution (voir le tableau 5.1).

Pour chaque algorithme, étant donné que l'on cherche à réaliser une carte peu dense, nous allons faire en sorte qu'un maximum de 20 points soient toujours visibles et reconnus. S'il en manque, suite à une divergence, un masquage trop long, ou un

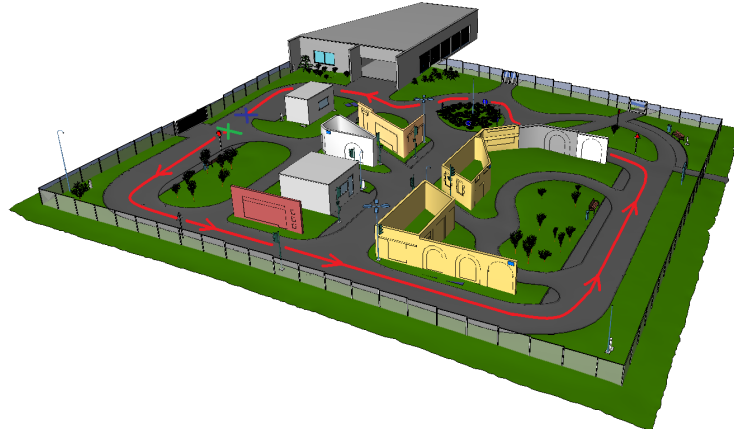


FIGURE 5.1 : Les croix verte et bleue symbolisent respectivement le début et la fin de la trajectoire. Le tracé en rouge montre le chemin suivi.

Nom	Correction du gain de Kalman	Correction des zones de recherche	Approximation d'un monde plan
A10	Non	Non	Non
A11	Non	Non	Oui
A12	Non	Oui	Oui
A13	Oui	Oui	Oui

TABLE 5.1 : Tableau récapitulatif des algorithmes de SLAM testés

résultat de corrélation trop faible, le processus essaiera d'en initialiser un nouveau tout en supprimant le point défaillant. Un test de convergence sera effectué afin de vérifier si le point ne doit pas être sauvegardé.

Dès lors que des points seront initialisés, les incertitudes 3D vont être suffisamment importantes pour que s'exprime l'intérêt de calculer correctement la projection de l'incertitude dans l'image. Cette remarque fait l'objet d'une première partie, préambule à l'analyse des différentes versions de l'algorithme de SLAM.

Puisqu'il s'agit de rejouer un trajet, c'est l'écart entre la trajectoire du premier véhicule (dite de référence) et la trajectoire rejouée qui importe : la vérité terrain telle qu'elle pourrait être donnée par un GPS-RTK n'est pas utile pour caractériser le système.

5.1.1 Définition des zones de recherche

Dans un premier temps, observons si le problème de la zone d'observation décelé théoriquement ainsi que dans la simulation persiste dans les expérimentations réelles. Pour cela, on effectue une simple recherche d'associations de points dans l'image en utilisant les données du GPS-RTK pour la position du véhicule, combinée à un modèle d'évolution pour estimer les faibles variations angulaires provoquées, lors de la conduite du véhicule, dans une zone droite de l'environnement PAVIN. Lors

de la première observation, un point est initialisé avec une incertitude importante sur le rayon de vue. Dans les images suivantes, on s'attache à trouver l'observation correspondant à notre estimation au sein de la zone définie, soit par la méthode des jacobiennes, soit par notre méthode. On obtient les résultats illustrés par la figure 5.2.

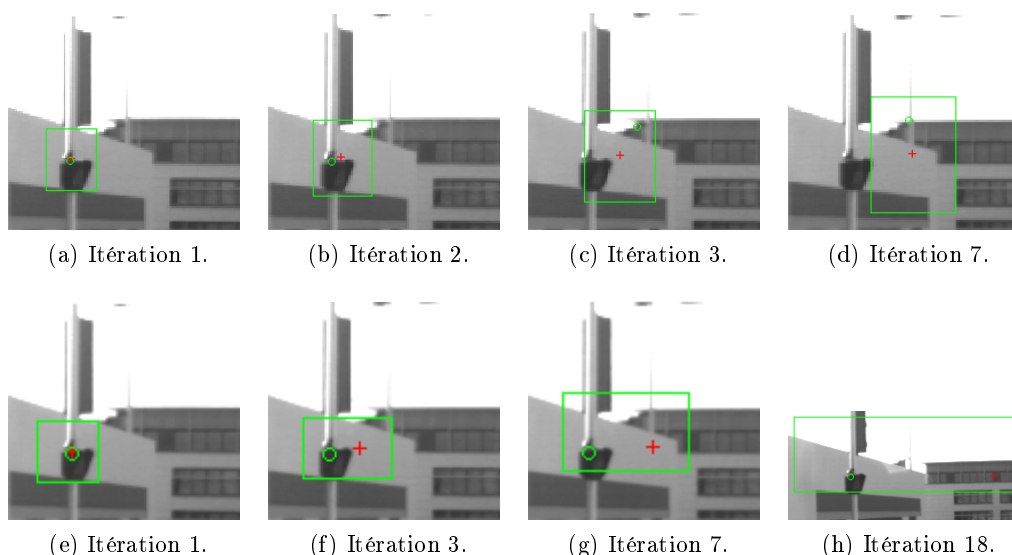


FIGURE 5.2 : La première ligne d'image présente les résultats de tracking, à diverses itérations, lorsque la zone de recherche est estimée via les jacobiennes. La deuxième ligne montre la zone définie par notre méthode.

Sur chaque figure, le cadre vert définit la zone de recherche correspondant à la projection de l'estimation d'un point 3D, symbolisée ici par une croix rouge. Si le maximum de corrélation du descripteur du point avec la zone de recherche est supérieur au seuil du ZNCC (ici défini à 0.8), une observation est associée et est représentée par un cercle vert. La première ligne d'image présente les résultats du tracking, à diverses itérations, lorsque la zone de recherche est estimée via les jacobiennes. On remarque que celle-ci n'est plus intègre et que, par conséquent, les observations ne sont plus cohérentes. Pire encore, les observations trouvées (dont le résultat de corrélation est supérieur au seuil) sont complètement fausses. La deuxième ligne montre la zone définie par notre méthode. On remarque que l'observation reste cohérente et ce, bien plus longtemps qu'avec la méthode précédente.

Finalement, le problème de la zone de recherche est bel et bien présent dans les expérimentations réelles.

Nous avons déjà montré dans la section 3.6 l'impact de la correction du gain de Kalman sur des données réelles. A présent, nous allons voir les effets de chaque amélioration dans un processus SLAM complet. Les prochaines sections développent chaque version de l'algorithme SLAM (A10, A11, A12 et A13) pour en faire une comparaison.

5.1.2 A10 : SLAM sans aucune amélioration

L'algorithme A10 est un SLAM qui initialise les points à une profondeur fixée à 500 mètres et constante sur l'axe de vue. Les zones de recherche dans les images sont calculées à partir des jacobiniennes et le gain du filtre de Kalman n'est en rien corrigé. Au cours de son traitement, il effectue 1506 initialisations pour obtenir un total de 51 points dans la carte. La figure 5.3 montre la trajectoire reconstruite ainsi que la carte correspondante.

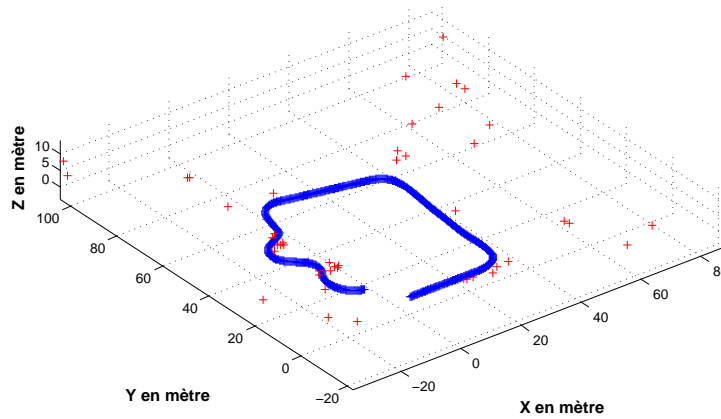


FIGURE 5.3 : Résultat de l'algorithme de SLAM sans aucune amélioration. Les points de la carte sont symbolisés par les croix rouges tandis que la trajectoire est représentée en bleu.

5.1.3 A11 : SLAM avec l'initialisation « monde plan »

L'algorithme A11 est un SLAM qui diffère de l'algorithme A10 par l'initialisation des points. En effet, on intègre à présent l'hypothèse « monde plan » qui offre, pour les points au sol, une meilleure initialisation (voir annexe D). Au cours de son traitement, 1275 initialisations sont réalisées et 172 points dans la carte ont suffisamment convergé pour être retenus. La figure 5.4 montre la trajectoire reconstruite ainsi que la carte correspondante.

5.1.4 A12 : SLAM avec l'initialisation « monde plan » et la correction des zones de recherche

Avec l'algorithme A12, on intègre la prise en compte de la non-linéarité des jacobiniennes dans la projection des zones d'incertitudes dans l'image. On retrouve l'amélioration présentée dans la section 2.3.3.3 du chapitre 2 concernant les zones de recherche d'association. Durant son déroulement, 973 initialisations sont réalisées tandis que 273 points dans la carte ont été retenus. La figure 5.5 montre la trajectoire reconstruite ainsi que la carte correspondante.

Notons au passage que, bien qu'un nombre conséquent de points aient été retenus, rares sont ceux qui se trouvent sur la trajectoire. Cela signifie qu'il sera difficile

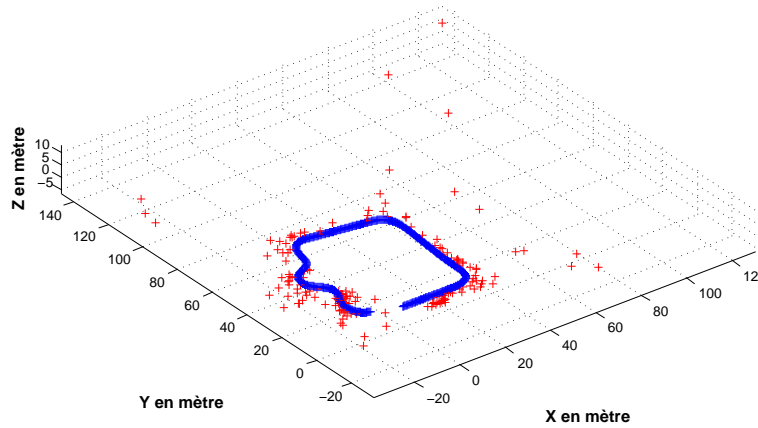


FIGURE 5.4 : Résultat de l'algorithme de SLAM avec l'initialisation « monde plan ». Les points de la carte sont symbolisés par les croix rouges tandis que la trajectoire est représentée en bleu.

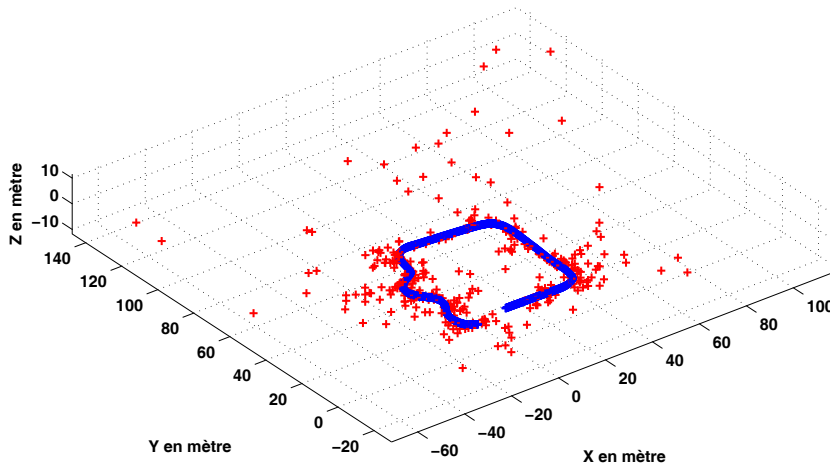


FIGURE 5.5 : Résultat de l'algorithme de SLAM avec l'initialisation « monde plan » et la correction des zones de recherche. Les points de la carte sont symbolisés par les croix rouges tandis que la trajectoire est représentée en bleu.

pour le véhicule de se localiser précisément en position. En effet, les points situés au loin permettent de corriger correctement l'orientation du véhicule mais ne donnent pas un positionnement précis. Ce problème sera résolu par l'approche intégrant la correction du gain de Kalman.

5.1.5 A13 : SLAM avec l'initialisation « monde plan », correction des zones de recherche et correction du gain de Kalman

Enfin, l'algorithme A13 intègre toutes les innovations présentées dans les chapitres précédents. Sont pris en compte les effets de la non-linéarité de la fonction

Algorithme	Nombre de points initialisés	Nombre de points dans la carte	Taille de la carte	Taille estimée au kilomètre
A10	1506	51	≈ 107 ko	≈ 627 ko
A11	1275	172	≈ 360 ko	$\approx 2,11$ Mo
A12	973	273	≈ 569 ko	≈ 3.42 Mo
A13	820	319	≈ 666 ko	$\approx 4,00$ Mo

TABLE 5.2 : Tableau comparatif des algorithmes SLAM.

d'observation sur la définition des zones de recherche dans l'image ainsi que sur le calcul du gain de Kalman, contraignant la mise à jour à ne pas diverger. Son déroulement laisse s'initialiser 820 points et 319 d'entre eux seront conservés dans la carte finale (qui deviendra par la suite la carte de référence). La figure 5.6 montre la trajectoire reconstruite ainsi que la carte correspondante.

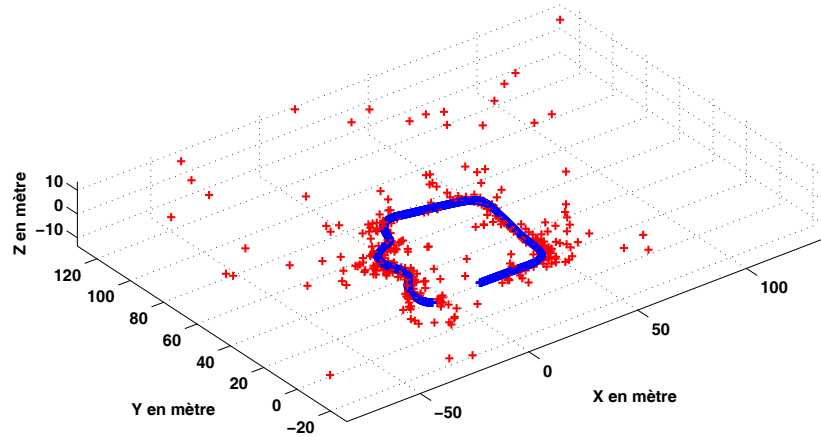


FIGURE 5.6 : Résultat de l'algorithme de SLAM avec l'initialisation « monde plan », correction des zones de recherche et correction du gain de Kalman. Les points de la carte sont symbolisés par les croix rouges tandis que la trajectoire est représentée en bleu.

Comme annoncé précédemment, on constate qu'un nombre important de points situés sur la trajectoire ont convergé. Ce résultat avait déjà été observé dans les figures 3.24a et 3.24b de la section 3.6.

5.1.6 Comparatif des algorithmes

On dresse une comparaison du nombre d'initialisations, du nombre de points retenus dans chaque carte et la taille correspondante en Mo.

Rappelons que la distance parcourue est de 170 mètres et que les descripteurs de chaque point sont de taille 15x15. On constate que chaque innovation permet d'accroître le nombre de points finalement conservés dans ce qui définira une carte de référence. Chaque point est défini par sa position (dans le repère monde), la

position de la première observation (dans le repère monde), les variances sur chaque axe (toujours dans le repère monde) ainsi que le descripteur. Exception faite pour les valeurs des pixels qui constituent le descripteur et qui sont codées sur 8 bits, toutes les autres informations sont codées sur 32 bits. La taille de la carte ainsi obtenue (≈ 4.4 Mo) est tout à fait raisonnable à l'échelle du kilomètre et justifie la dénomination « peu dense ». De plus, cet écart à la contrainte (10 Mo) nous permet d'envisager une localisation tout aussi précise dans des environnements moins favorables. Le nombre de points toujours observables par image, 20 dans notre cas, pourrait être augmenté pour répondre à cette problématique.

Lors de l'exécution de l'algorithme A13, 1016 corrections du gain de Kalman ont été réalisées et la plus importante a été de 0.0024656. Cela signifie qu'en considérant la mise à jour de Kalman sans correction à hauteur de 100%, la non-linéarité représentait à elle seule dans ce cas 99.9975% de la mise à jour initiale ! Cette correction apparaît donc comme indispensable dès lors que l'observation est précise tandis que la méconnaissance de l'objet estimé est importante. Un autre cas de figure renforce l'utilisation de cette correction : l'augmentation de la vitesse du véhicule. En effet, la fréquence de la caméra est fixe et l'effet de la non-linéarité est directement lié à l'innovation dans l'image. Par conséquent, donc plus le véhicule va vite, plus l'erreur de linéarisation est importante, plus l'innovation est grande et plus la mise à jour diverge.

Construire une carte n'est pas la seule partie de notre problématique, encore faut-il pouvoir évoluer et se localiser sans sortir du couloir de navigation. Autrement dit, il faut que la carte construite permette à un véhicule, exécutant l'algorithme de localisation, de rejouer la même trajectoire en restant sous la contrainte des 10 cm d'écart : ce point est abordé dans la section suivante.

5.2 Localisation à partir d'une carte de référence sur le site PAVIN

La section précédente nous fournit, avec les résultats de l'algorithme A13, une carte de référence et une trajectoire à suivre. Un deuxième véhicule peut alors rejouer le chemin en naviguant et en se localisant dans celle-ci. En reprenant le même jeu d'image et de données proprioceptives, on peut rejouer le chemin et vérifier que l'on se localise bien dans le couloir de navigation défini à 10 cm de part et d'autre de la trajectoire de référence. La figure 5.7 montre la superposition de la trajectoire de référence avec celle estimée lors du second passage. Le figure 5.8 montre que l'écart de la localisation lors du second passage par rapport à la trajectoire de référence reste en dessous de 8 cm ce qui vérifie ici la contrainte de positionnement imposée.

5.3 SLAM avec carte de référence

Nous avons réussi à développer un algorithme SLAM permettant de construire une carte dont la densité est faible (modulable en cas d'insuffisance d'information)

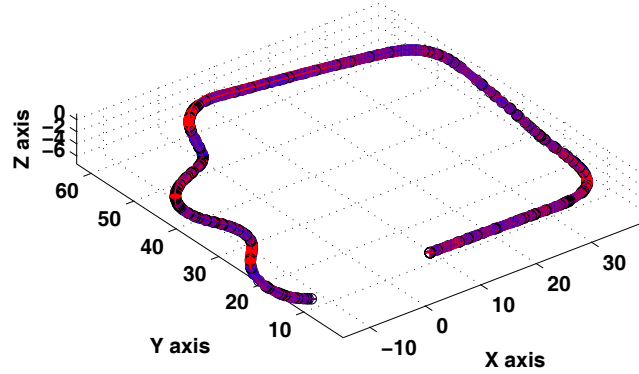


FIGURE 5.7 : La trajectoire de localisation (en rouge) est très similaire à la trajectoire de référence (en bleu).

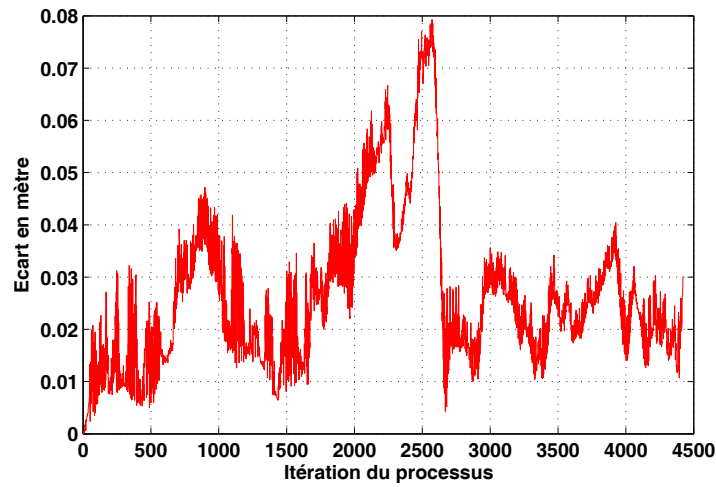


FIGURE 5.8 : Ecart de chaque résultat de localisation par rapport à la trajectoire de référence (distance euclidienne).

et un algorithme de localisation permettant de naviguer dans une carte de référence avec un couloir de navigation (défini à 10 cm de part et d'autre d'une trajectoire de référence).

A présent, nous allons montrer l'application avec la prise en compte du biais sur des données réelles qui permet la cohabitation des deux cartes (de référence et SLAM). Pour cela, il faut amener le véhicule dans une zone qui n'a pas encore été cartographiée et qui n'appartient donc pas à la trajectoire de référence. La figure 5.9 montre le chemin suivi.

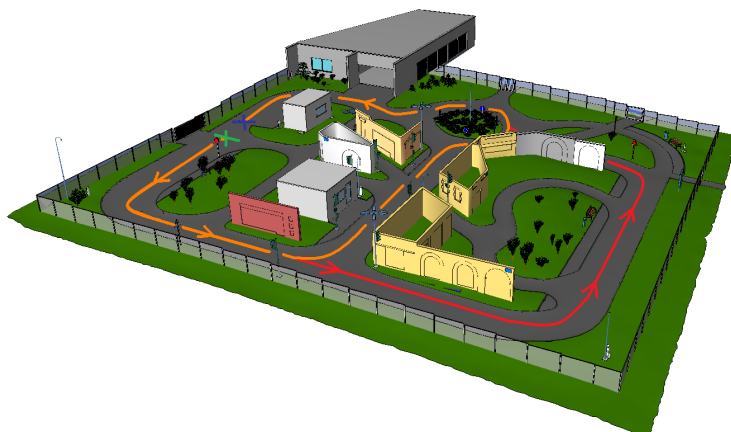


FIGURE 5.9 : Les croix verte et bleue symbolisent respectivement le début et la fin de la trajectoire. Le tracé en orange montre la trajectoire suivie et son écart par rapport à la trajectoire de référence en rouge.

Pour montrer l'intérêt de la prise en compte de la dérive du SLAM, nous allons réaliser deux versions de rejeu sur la nouvelle trajectoire. La première se fera sans prise en compte du biais tandis qu'il sera estimé dans une deuxième version. Les figures 5.10 et 5.11 montrent le résultat de localisation de chaque variante.

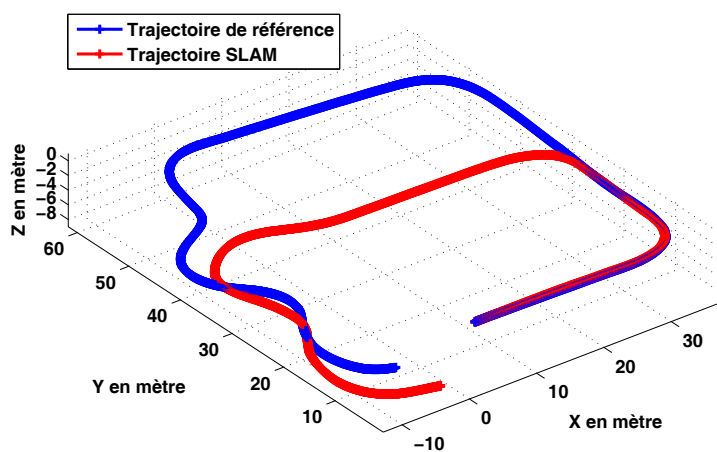


FIGURE 5.10 : Résultat de localisation (en rouge) par rapport à la trajectoire de référence (en bleu) sans prise en compte du biais entre la carte de référence et la carte SLAM. On constate que l'algorithme n'arrive pas à se re-localiser dans la carte de référence (la trajectoire verte quitte la trajectoire rouge pour se confondre avec la bleue).

Les variances du biais permettent de visualiser trois phases :

- le véhicule circule dans sa carte de référence (sur la trajectoire de référence) ;

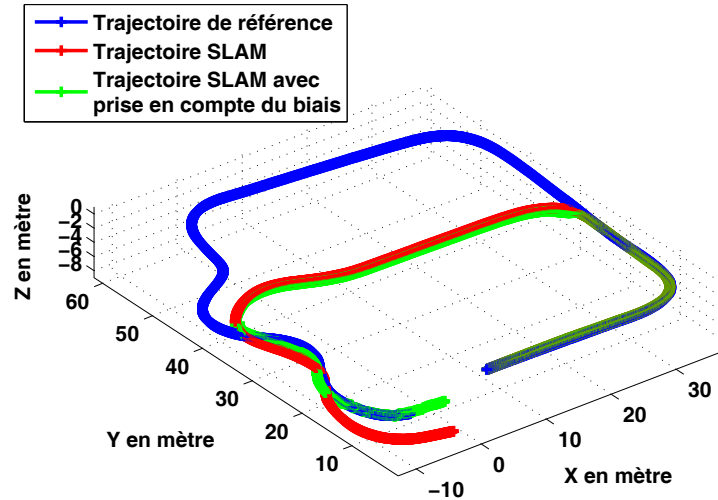


FIGURE 5.11 : Résultat de localisation (en rouge) par rapport à la trajectoire de référence (en bleu). La courbe verte montre le résultat de la trajectoire en tenant compte de l'estimation du biais. On constate que l'algorithme réussit à se re-localiser dans la carte de référence.

- le véhicule SLAM sans la carte de référence et l'incertitude du biais augmente (hors de la trajectoire de référence) ;
- le véhicule reconnaît les amers de sa carte de référence et reprend sa navigation sans la carte SLAM. L'incertitude du biais est mise à jour et redevient presque nulle (retour sur la trajectoire de référence).

La figure 5.12 illustre le propos en précisant chacune des phases sur les courbes d'évolution des variances des paramètres du biais.

Pour chacune des deux variantes, l'écart à la trajectoire de référence peut être calculé (voir figure 5.13). On retrouve en plus les trois phases présentées précédemment. On remarque une augmentation en toute fin : celle-ci est due au fait que le jeu de données utilisé pour générer les informations de référence est plus court que celui du SLAM. Cette limite est symbolisée dans le figure 5.13 par une ligne magenta.

5.4 Conclusion générale des expérimentations

En conclusion de cette partie, nous venons de voir que la redéfinition des zones de recherche par une projection plus juste de l'ellipsoïde d'incertitude donne de meilleurs résultats dans la recherche d'observations (correspondances dans l'image d'un descripteur). De plus, la correction du gain de Kalman réduit de manière drastique l'apparition de divergence.

Il reste cependant un détail inhérent au processus SLAM : son biais (c'est à dire sa dérive). En effet, rappelons que nous travaillons ici avec des cartes peu denses (350 points pour 170 mètres parcourus). La fermeture de boucle, contrainte non imposée dans le cadre du projet, est la condition sine qua non de l'intégrité du SLAM. En

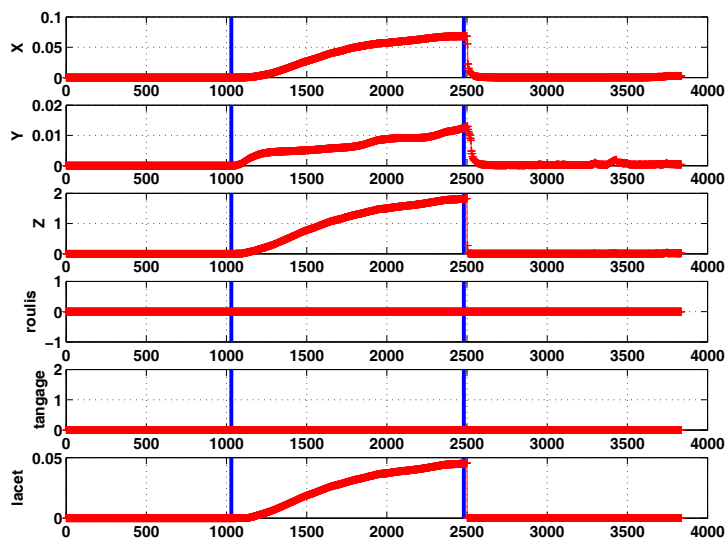


FIGURE 5.12 : Les courbes des variances du biais ont été compartimentées pour mieux visualiser les trois phases : la circulation dans la carte de référence (en premier à gauche), la navigation sans la carte de référence (en second au milieu) et le retour dans la carte de référence (en dernier sur la droite). Les variances sur le roulis et le tangage sont trop faibles pour observer une variation sur l'intervalle où le véhicule dévie de la trajectoire de référence.

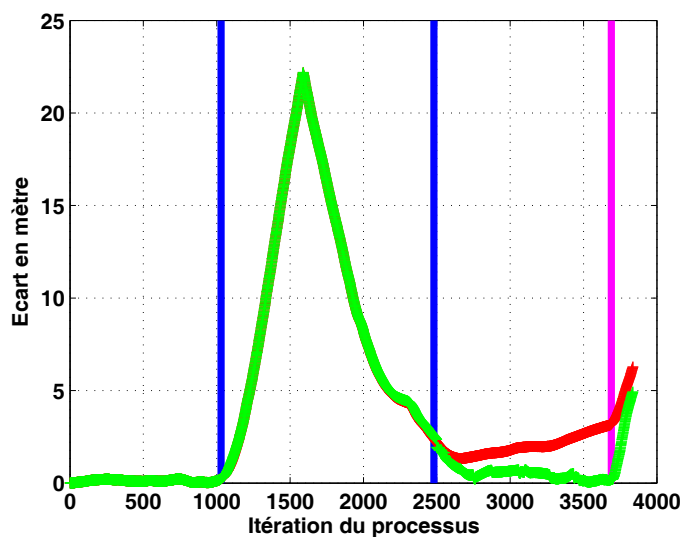


FIGURE 5.13 : La trajectoire de référence est comparée, par un calcul de distance euclidienne, à la trajectoire du SLAM avec prise en compte du biais (en vert) et sans (en rouge). Les lignes bleues verticales permettent de différencier les trois phases de navigation et la ligne magenta montre la fin de la trajectoire de référence. On constate que la mise à jour des paramètres du biais permet de réduire l'écart.

l'absence de bouclage, nous avons montré que la prise en compte du biais permet de maintenir la cohérence entre la carte de référence et la carte SLAM, cohérence qui a été garantie suite à une déviation de plus de 20 mètres. Enfin, rappelons que l'algorithme de localisation permet de naviguer précisément dans une carte peu dense et en temps réel (7 Hz sous MATLAB).

En préambule de ce chapitre, nous avons vérifié que la projection de l'ellipsoïde d'incertitude par la méthode des jacobiennes était trop restrictive et ne permettait plus de faire correspondre une observation à un point toujours observable. Ensuite, nous avons montré que notre méthode de calcul offrait de meilleurs résultats.

Suite à cette validation, nous avons ajouté les améliorations une à une sur un parcours défini sur le site PAVIN. Les résultats de chaque version ont été rassemblés dans un tableau pour en faire une synthèse, mettant en avant l'intérêt de chaque apport.

Enfin, la prise en compte de la dérive entre deux cartes réalisées par deux processus indépendants permet d'élaborer des trajectoires plus complexes et moins dépendantes de la trajectoire de référence. Ce dernier algorithme autorise un véhicule à s'éloigner de la trajectoire de référence, voire à explorer des zones n'appartenant pas à la carte de référence. Le retour dans la carte de référence permet de reprendre la mission originelle : rejouer une trajectoire prédéfinie.

Conclusions et perspectives

Conclusions

Le projet de thèse énoncé lors de l'introduction proposait, dans un premier temps, de faire en sorte qu'un véhicule puisse évoluer dans un couloir de navigation défini autour d'une trajectoire dite de référence avec un écart latéral contraint à 10 cm. Le système se devait suffisamment discret pour ne pas être détecté et suffisamment simple, en terme de capteur, pour pouvoir être implémenté dans différents types de véhicule. De plus, son coût devait être faible et son exécution temps réel.

Il est évident que, sans prendre d'information sur l'environnement, nous aurions été forcés de naviguer en n'utilisant que les données proprioceptives du véhicule. Cela n'est pas réalisable en l'état compte tenu des contraintes de coût imposées. Par conséquent, ces informations de l'environnement ont pris la dénomination de carte et le projet nous laissait maître de sa définition ainsi que de sa constitution. Cependant, cette carte devait être légère, ce qui signifie une quantité de données faible au kilomètre. En ce sens, le projet proposait une contrainte de 10 Mo par kilomètre. Nous sommes arrivés à réaliser une carte dont la taille approchait à peine les 5 Mo par kilomètre. Ainsi, on peut espérer que, dans des environnements moins favorables, la contrainte de 10 Mo suffira à générer une carte suffisamment informative pour garantir la précision de notre localisation. Cette augmentation se traduira par la recherche d'un plus grand nombre de points dans chaque image (dans notre approche, ce nombre était de 20).

Dans un deuxième temps, le projet envisage un enrichissement de la carte par la création d'une carte secondaire, propre au véhicule qui rejoue le chemin (la carte SLAM). En effet, le véhicule est susceptible de s'éloigner de la trajectoire référence pour contourner un obstacle, absent par exemple lors de la création de la trajectoire de référence. Dès lors, le véhicule évolue dans un univers inconnu et doit par conséquent, cartographier ce nouvel environnement afin de préciser sa localisation. De plus, il doit pouvoir reprendre la trajectoire de initiale et reconnaître sa position dans la carte de référence. Il a été constaté que l'algorithme de SLAM, utilisé par exemple par un véhicule téléopéré, pourrait tout à fait permettre la construction des données de référence. Cette possibilité se devait d'être étudiée et faisait partie du second volet du projet de thèse.

Au cours de l'élaboration d'une solution pour chacune des problématiques, nous avons révélé des problèmes liés à la non-linéarité de la fonction d'observation. Celle-ci se traduit à plusieurs niveaux et nous allons revenir successivement sur chacun de ces points.

Tout d'abord, revenons sur notre **première contribution** : la définition des zones de recherche dans l'image. Traditionnellement, cette zone est calculée en projetant l'incertitude du point et de la position de l'observateur dans le plan image à l'aide des jacobiennes du modèle d'observation. Si le modèle est non-linéaire, les

jacobiennes correspondent aux linéarisations au premier ordre. Par conséquent, elles peuvent parfois être incohérentes voire trop restrictives. Nous avons présenté une méthode permettant d'extraire une zone rectangulaire dans l'image qui contient parfaitement la projection des incertitudes. Si le point 3D cherché est intègre vis-à-vis de son estimation, alors nous garantissons que son projeté se trouve dans la zone de recherche.

Notre **deuxième contribution** traduit le pendant de cette méthode : la redéfinition du processus d'initialisation des points 3D au sein du SLAM. Généralement, cette initialisation est réalisée par le filtre de Kalman. Nous avons montré que cette manière de faire n'était pas cohérente avec l'observation de l'environnement et nous avons proposé une nouvelle méthode. Comme nous avons calculé la projection exacte d'un ellipsoïde d'incertitude dans l'espace 3D, notre méthode d'initialisation cherche l'incertitude qui, par la même méthode, donne une erreur définie dans l'image. Dès lors, que ce soit pour la création de l'incertitude 3D associée à un point ou pour la projection de celle-ci dans l'espace image, nos méthodes s'affranchissent complètement du calcul des jacobiennes du modèle d'observation, limitant les effets indésirables d'une mauvaise linéarisation.

Enfin, notre **troisième contribution** se situe au niveau de l'expression de la non-linéarité dans l'exécution du filtre de Kalman et plus particulièrement dans le calcul du gain. Nous avons vu, sur un cas simple, comment une seule mise à jour peut faire perdre l'intégrité sur la pose d'un point. Nous avons abordé ce problème sous un angle généraliste si bien que notre méthode n'est pas limitée au cadre du SLAM visuel. Les résultats montrent que la correction du gain de Kalman proposée permet de conserver l'intégrité de manière plus fiable et limite l'expression de la non-linéarité du modèle. Les points sont alors observés sur un plus grand nombre d'images et leur convergence est bien meilleure. Par ailleurs, l'expression de la non-linéarité est d'autant plus importante que l'estimation est loin de l'observation dans l'image (cela traduit une innovation importante) comme cela peut être le cas pour des vitesses plus importantes. Notre méthode, tout en contrôlant l'expression de la non-linéarité, permet donc aux véhicules de se déplacer plus vite pour une fréquence de caméra donnée.

Ces trois principales avancées ont été implémentées dans différentes approches pour atteindre les objectifs du projet. Deux fonctions de localisation ont été développées en premier lieu :

- l'une s'appuyant sur un a priori précis de la localisation en plus des différentes données issues des capteurs ;
- l'autre fondée sur une connaissance plus imprécise de la position.

La première est une localisation dite en ligne car utilisée tout au long de la mission. Son exécution est temps réel et l'incertitude mesurée en permanence est un bon indicateur de la précision de la localisation fournie. La seconde, plus longue car plus complexe dans son exécution, est réservée au cas de la pose initiale du véhicule. En effet, les zones de recherche dans l'image étant grandes (impact direct de l'imprécision de la connaissance de la pose a priori), le cas de mauvaise association de données devient alors un problème majeur.

Pour chacun des deux algorithmes, nous avons présenté des expérimentations sur des données simulées dans un premier temps puis réelles dans un second temps en démontrant que le véhicule respecte la contrainte d'écart latéral tout au long de la trajectoire de référence.

Dans ce premier volet du projet de thèse, nous possédions la carte et la trajectoire de référence. Dans un second temps, l'élaboration d'un algorithme SLAM permettra à un véhicule de s'éloigner de la trajectoire de référence pour y revenir par la suite. La création d'une nouvelle carte propre au véhicule permet de poursuivre le processus de localisation dans un environnement qui n'a pas été exploré lors de l'élaboration de la carte de référence. Pouvoir mêler cette nouvelle carte, identifiée comme la carte SLAM, avec la carte de référence ne peut se faire directement à cause de la dérive du processus SLAM. Pour contenir l'intégrité et permettre au véhicule de remettre en correspondance ses observations avec la carte de référence, nous avons introduit un biais dont les paramètres évoluent dans le vecteur d'état comme ceux du véhicule. Une nouvelle fois, nous avons éprouvé notre méthode sur des données de simulation puis sur des données réelles.

En parallèle, l'élaboration d'un processus SLAM nous laissait percevoir un moyen de générer la carte de référence. Pourquoi un véhicule ne pourrait-il pas créer une carte SLAM qui deviendrait une carte de référence pour ceux qui souhaiteraient rejouer la trajectoire ? Cette problématique est adressée dans les résultats sur données réelles. Nous avons montré qu'il était tout à fait possible de construire une carte d'une densité suffisamment faible pour respecter la contrainte de volume d'information et d'une pertinence suffisamment importante pour assurer un rejeu dans le couloir de navigation défini à 10 cm autour de la trajectoire de référence.

Enfin, nous avons abordé simplement le cas bi-monoculaire en montrant une estimation plus rapide des points observés que dans le cas monoculaire. Les améliorations ont ici aussi été implémentées et nous avons constaté que le gain n'était pas significatif en terme de localisation. Cependant, cette méthode est tout à fait pertinente pour la calibration extrinsèque d'un banc stéréoscopique. Mieux encore, l'utilisation de l'approche bi-monoculaire permet une actualisation constante de la position relative des deux caméras et offre une solution élégante au problème de perte de calibration stéréoscopique due principalement aux reliefs accidentés du terrain.

Perspectives

Les travaux présentés dans ce manuscrit permettent de réaliser une fonction de localisation précise à partir d'une carte de référence. Nous proposons également une méthode de construction de la carte permettant, à terme, la réalisation d'un convoi par exemple.

Une hypothèse sous-jacente aux travaux présentés est celle d'un environnement statique. Il s'agit là d'une contrainte forte qui pourrait mettre en échec le processus de localisation. Une solution serait d'utiliser l'approche bi-monoculaire pour déterminer les points se déplaçant dans l'image de manière incohérente en comparaison

avec leur estimation. De nombreux travaux traitent de cette problématique qui est encore d'actualité et se place bien souvent en amont des algorithmes de localisation traditionnels.

D'un autre côté, une dernière étape serait nécessaire pour obtenir un système autonome : définir et implémenter une loi de commande. Les données réelles ont été acquises par un véhicule téléopéré et malheureusement nous n'avons pas eu le temps de faire en sorte que l'algorithme soit la fonction de localisation d'un processus de mobilité. Cette ultime étape permettra à un véhicule de se mouvoir de manière autonome en garantissant un écart latéral à la trajectoire de référence faible.

L'algorithme SLAM développé donne des angles d'approches innovants en ce qui concerne, d'une part, le problème d'initialisation des points 3D, et d'autre part, le problème de la non-linéarité du filtre de Kalman étendu. Ces améliorations, comme mentionné précédemment, ne sont pas restreintes au Visual-SLAM et il aurait été intéressant de comparer les résultats de SLAM autres que visuels en y intégrant tout simplement les modifications présentées.

Toujours en relation avec les points 3D, leur stratégie de sélection dans la carte de référence s'appuie sur une étude empirique renforcée par l'état de l'art. Cependant, il serait plus judicieux de choisir les points en fonction de la dimension à corriger car jugée trop incertaine. Ce choix ne devra être dépendant que de la position du point 3D et devra se faire en ayant établi quelle dimension est prioritaire dans le processus de mise à jour. Nous avons montré combien ce problème est complexe et que, dans une approche par focalisation, pouvoir faire ce choix permettra un gain en temps de calcul ainsi qu'en précision.

D'un point de vue théorique, nous avons montré une façon d'exprimer la projection d'une incertitude 3D dans une image. Cette projection est une ellipse dont nous avons donné une manière de calculer la forme quadratique associée. Cette ellipse représente, dans l'image, l'incertitude a posteriori notée dans le filtre de Kalman $\mathbf{H}_{k+1}\mathbf{P}_{k+1|k}\mathbf{H}_{k+1}^T$. Or elle n'a pas été utilisée de manière optimale. En effet, nous nous sommes arrêtés au calcul d'une boîte rectangulaire englobant cette ellipse. Il serait intéressant de poursuivre le calcul jusqu'aux paramètres de l'ellipse, la matrice de sa forme quadratique, afin d'intégrer cette incertitude a posteriori dans l'approche de Kalman utilisée. Cependant, cette incertitude n'est plus centrée sur le maximum de vraisemblance, à savoir le projeté du point 3D ce qui complique le calcul de l'innovation. Il reste donc une étape à découvrir pour pouvoir tenir compte de cette ellipse, projection exacte d'un ellipsoïde d'incertitude.

Par ailleurs, le modèle du biais introduit pour compenser la dérive du processus SLAM n'a pas été développée finement. Cette dérive, non reproductible rappelons le, est liée aux capteurs, à l'environnement ainsi qu'au véhicule lui-même. Développer une méthode de caractérisation, comme cela a pu être le cas pour le modèle de bruit associé au modèle d'évolution, permettrait au système d'être robuste aux différentes éventualités de contournement d'obstacle.

Finalement, il reste un point important en rapport direct avec la notion de décision : comment faire confiance à un robot ? Cette question tente de trouver une réponse dans l'évaluation de l'incertitude associée à une estimation donnée. Mais

alors, comment garantir l'intégrité ? L'humain n'est pas infaillible et l'existence du pouvoir judiciaire en est la preuve. Nous sommes capables de faire confiance à une machine dès lors qu'elle ne met pas directement notre vie en danger. De manière générale, on se sent rassurer si, à tout moment, on peut voir venir un danger et s'en prémunir avant qu'il ne nous atteigne. Comment se prémunir à bord d'un véhicule autonome ? Cette question reste ouverte et montre que les perspectives sont encore nombreuses pour élaborer un processus complet, quel que soit la finalité du projet.

Pour conclure, ce projet nous a permis de travailler sur les nombreux problèmes du SLAM visuel et du filtre de Kalman. Les solutions apportées restent novatrices et argumentées aussi bien d'un point de vue scientifique que du point de vue de l'état de l'art. Cependant, il reste encore de nombreux problèmes dont les solutions permettront sans doute un jour de se rendre d'un point A à un point B de manière complètement autonome. Nous navigueront alors à bord d'un engin entièrement robotisé, pouvant tenir compte d'autant de paramètres extérieurs et environnementaux que n'importe quel conducteur humain.

Le lieu des points d'un ellipsoïde n'englobant pas l'origine et dont les plans tangents contiennent l'origine

On se propose ici de montrer que le lieu des points d'un ellipsoïde n'englobant pas l'origine et dont les plans tangents contiennent l'origine est une ellipse. Un lemme de commutativité matricielle est nécessaire pour établir le système de relations qui conduisent à la solution.

A.1 Lemme de commutativité matricielle

Nous allons, dans cette partie rappeler et démontrer un résultat d'algèbre linéaire utile pour la partie suivante à savoir que pour toute matrice $\mathbf{M} \in \mathcal{GL}_n(\mathfrak{R})$ symétrique, et \vec{U} et \vec{V} des vecteurs de taille $[n \times 1]$, on a :

$$\vec{U}^T \mathbf{M} \vec{V} = \vec{V}^T \mathbf{M} \vec{U} \quad (\text{A.1})$$

Pour cela, décomposons chaque multiplication et travaillons sur les sommes :

$$\vec{V}^T \mathbf{M} \vec{U} \Rightarrow \alpha = \sum_{k=1}^n \left(\sum_{j=1}^n v_{1j} m_{jk} \right) u_{k1} \quad (\text{A.2})$$

$$\vec{U}^T \mathbf{M} \vec{V} \Rightarrow \alpha = \sum_{k=1}^n \left(\sum_{j=1}^n u_{1j} m_{jk} \right) v_{k1} \quad (\text{A.3})$$

avec m_{ij} l'élément de \mathbf{M} situé ligne i et colonne j . Il en est de même pour u_{k1} et v_{k1} , respectivement le $k^{\text{ième}}$ élément de \vec{U} et de \vec{V} . Menons les développements de chaque relation de front et montrons qu'elles sont équivalentes :

$$\begin{aligned} & \sum_{k=1}^n \left(\sum_{j=1}^n v_{1j} m_{jk} \right) u_{k1} & \sum_{k=1}^n \left(\sum_{j=1}^n u_{1j} m_{jk} \right) v_{k1} \\ \Leftrightarrow & \sum_{k=1}^n \sum_{j=1}^n v_{1j} m_{jk} u_{k1} & \sum_{k=1}^n \sum_{j=1}^n u_{1j} m_{jk} v_{k1} \\ \Leftrightarrow & \sum_{k=1}^n \sum_{j=1}^n v_{1j} m_{kj} u_{k1} & \sum_{k=1}^n \sum_{j=1}^n u_{1j} m_{kj} v_{k1} \\ \Leftrightarrow & \sum_{j=1}^n \sum_{k=1}^n u_{1k} m_{kj} v_{j1} & \sum_{j=1}^n \sum_{k=1}^n u_{1j} m_{jk} v_{k1} \\ \Leftrightarrow & \sum_{k=1}^n \sum_{j=1}^n u_{1j} m_{jk} v_{j1} & \sum_{k=1}^n \sum_{j=1}^n u_{1j} m_{jk} v_{k1} \end{aligned}$$

A.2 Le lieu des points de l'ellipsoïde dont les plans tangents contiennent l'origine est une ellipse

Soit M la matrice symétrique réelle de variance-covariance définie comme suit :

$$\begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix}$$

Cette matrice définit une forme quadratique dans l'espace 3D et définit un ellipsoïde orienté. Une des relations vérifiées par les points de l'ellipsoïde \mathcal{E} engendré par M est la distance de Mahalanobis :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathcal{E} \Leftrightarrow \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}^T M^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 1 \quad (\text{A.4})$$

De plus, nous cherchons les points dont les plans tangents passent par l'origine. Se posent alors deux questions :

- Quels sont les plans tangents à \mathcal{E} ?
- Quels sont ceux qui contiennent l'origine ?

A.2.1 Plan tangent à \mathcal{E}

La normale d'une surface en un point est portée par le gradient en ce point :

$$\vec{n} = \overrightarrow{\text{grad}}\mathcal{E}(x, y, z) = \left(\frac{\partial \mathcal{E}}{\partial x}, \frac{\partial \mathcal{E}}{\partial y}, \frac{\partial \mathcal{E}}{\partial z} \right)^T \quad (\text{A.5})$$

L'inverse de M se calcule littéralement :

$$M = \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix} \Leftrightarrow M^{-1} = \begin{pmatrix} A & B & C \\ B & D & E \\ C & E & F \end{pmatrix} \quad (\text{A.6})$$

$$\begin{aligned} A &= (df - e^2)/\det M & D &= (af - c^2)/\det M \\ B &= (ce - bf)/\det M & E &= (cb - ae)/\det M \\ C &= (be - cd)/\det M & F &= (ad - b^2)/\det M \end{aligned}$$

Par conséquent, on peut écrire \vec{n} sous la forme suivante :

$$\vec{n} \equiv \begin{pmatrix} (x - x_0)A + (y - y_0)B + (z - z_0)C \\ (x - x_0)B + (y - y_0)D + (z - z_0)E \\ (x - x_0)C + (y - y_0)E + (z - z_0)F \end{pmatrix} \quad (\text{A.7})$$

On sait que la normale à la surface en $(x, y, z)^T$ est portée par \vec{n} . Par conséquent, le plan tangent est orthogonal à \vec{n} .

Les points appartenant au plan vérifient donc la relation suivante :

$$\begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} \in P_{\perp \vec{n}} \Leftrightarrow (x_p - x, y_p - y, z_p - z) \cdot \vec{n} = 0 \quad (\text{A.8})$$

A.2.2 Plan tangent à \mathcal{E} passant par l'origine

Le point $(0, 0, 0)^T$ vérifie l'équation des plans tangents cherchés :

$$\begin{aligned} & (x, y, z) \cdot \vec{n} = 0 \\ \Leftrightarrow & (x, y, z) \cdot \begin{pmatrix} (x - x_0)A + (y - y_0)B + (z - z_0)C \\ (x - x_0)B + (y - y_0)D + (z - z_0)E \\ (x - x_0)C + (y - y_0)E + (z - z_0)F \end{pmatrix} = 0 \end{aligned} \quad (\text{A.9})$$

qui peut être mis sous la forme :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}^T M^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 0 \quad (\text{A.10})$$

A présent, on possède les deux relations permettant de constituer le système d'équations du problème. Sachant que les points appartiennent à un ellipsoïde centré sur $(x_0, y_0, z_0)^T$ et à ses plans tangents passant par l'origine, on obtient :

$$\begin{cases} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}^T M^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 1 \\ \begin{pmatrix} x \\ y \\ z \end{pmatrix}^T M^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 0 \end{cases} \quad (\text{A.11})$$

En combinant les deux relations et en utilisant le lemme démontré précédemment, on peut ré-écrire la système sous la forme suivante :

$$\begin{cases} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}^T M^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 1 \\ \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}^T M^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = -1 \end{cases} \quad (\text{A.12})$$

**Annexe A. Le lieu des points d'un ellipsoïde n'englobant pas l'origine
140 et dont les plans tangents contiennent l'origine**

Sous cette forme, on constate clairement que les points de coordonnées (x, y, z) cherchés se trouvent sur l'intersection d'un ellipsoïde (première relation) avec un plan (deuxième relation). Or tout plan qui coupe un ellipsoïde définit alors une ellipse. Par conséquent, le lieu des points d'un ellipsoïde n'englobant pas l'origine, dont les plans tangents contiennent l'origine, est une ellipse.

Projection d'un ellipsoïde dans une image : calcul des tangentes limites

B.1 Introduction

Etant donné que la projection d'un ellipsoïde d'incertitude dans une image par la méthode des jacobienues ne donne pas un résultat cohérent, on se propose ici de calculer 4 points particuliers permettant de définir parfaitement l'enveloppe de la projection dans l'image, des points définis sur l'ellipsoïde 3D.

B.2 Equation principale du problème

Soit M la matrice symétrique réelle de variance-covariance définie comme suit :

$$\begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix}$$

Cette matrice définit une forme quadratique dans l'espace 3D et définit un ellipsoïde orienté. Une des relations vérifiées par les points de l'ellipsoïde \mathcal{E} engendré par M est la distance de Mahalanobis :

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathcal{E} \Leftrightarrow \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}^T M^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 1 \quad (\text{B.1})$$

De plus, nous cherchons les points dont les plans tangents passent par l'origine. Se posent alors deux questions :

- Quels sont les plans tangents à \mathcal{E} ?
- Quels sont ceux qui contiennent l'origine ?

B.2.1 Plan tangent à \mathcal{E}

La normale d'une surface en un point est portée par le gradient en ce point :

$$\vec{n} = \overrightarrow{\text{grad}}\mathcal{E}(x, y, z) = \left(\frac{\partial \mathcal{E}}{\partial x}, \frac{\partial \mathcal{E}}{\partial y}, \frac{\partial \mathcal{E}}{\partial z} \right)^T \quad (\text{B.2})$$

L'inverse de M se calcule littéralement :

$$M = \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix} \Leftrightarrow M^{-1} = \begin{pmatrix} A & B & C \\ B & D & E \\ C & E & F \end{pmatrix} \quad (\text{B.3})$$

$$\begin{aligned} A &= (df - e^2)/\det M & D &= (af - c^2)/\det M \\ B &= (ce - bf)/\det M & E &= (cb - ae)/\det M \\ C &= (be - cd)/\det M & F &= (ad - b^2)/\det M \end{aligned}$$

Par conséquent, on peut écrire \vec{n} sous la forme suivante :

$$\vec{n} \equiv \begin{pmatrix} (x - x_0)A + (y - y_0)B + (z - z_0)C \\ (x - x_0)B + (y - y_0)D + (z - z_0)E \\ (x - x_0)C + (y - y_0)E + (z - z_0)F \end{pmatrix} \quad (\text{B.4})$$

On sait que la normale à la surface en $(x, y, z)^T$ est portée par \vec{n} . Par conséquent, le plan tangent est orthogonal à \vec{n} .

Les points appartenant au plan vérifient donc la relation suivante :

$$\begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} \in P_{\perp \vec{n}} \Leftrightarrow (x_p - x, \quad y_p - y, \quad z_p - z) \cdot \vec{n} = 0 \quad (\text{B.5})$$

B.2.2 Plan tangent à \mathcal{E} passant par l'origine

Le point $(0, 0, 0)^T$ vérifie l'équation des plans tangents cherchés :

$$\begin{aligned} &(x, \quad y, \quad z) \cdot \vec{n} = 0 \\ \Leftrightarrow &(x, \quad y, \quad z) \cdot \begin{pmatrix} (x - x_0)A + (y - y_0)B + (z - z_0)C \\ (x - x_0)B + (y - y_0)D + (z - z_0)E \\ (x - x_0)C + (y - y_0)E + (z - z_0)F \end{pmatrix} \end{aligned} \quad (\text{B.6})$$

B.2.3 Recherche des plans limites

Il faut chercher les plans tangents à \mathcal{E} , contenant l'origine et dont l'intersection avec le plan image donne soit une droite horizontale dans l'espace 3D, soit une droite verticale.

Il serait possible de calculer l'intersection des plans tangents avec le plan image et de discuter selon l'orientation des droites résultantes de l'intersection.

Cependant, on peut aussi constater que, comme les plans générant des droites verticales sont eux-même verticaux et qu'ils incluent l'origine du repère, alors ils contiennent l'axe \vec{z} tout entier. Cela se traduit par la relation suivante :

$$\vec{z} \cdot \vec{n} = 0 \quad (\text{B.7})$$

Or $\vec{z} = (0, 0, 1)^T$, par conséquent :

$$\vec{z} \cdot \vec{n} = 0 \Leftrightarrow (x - x_0)C + (y - y_0)E + (z - z_0)F = 0 \quad (\text{B.8})$$

Il en est de même pour les plans horizontaux qui contiennent l'axe \vec{y} . Il s'en suit la relation suivante :

$$\vec{y} \cdot \vec{n} = 0 \Leftrightarrow (x - x_0)B + (y - y_0)D + (z - z_0)E = 0 \quad (\text{B.9})$$

B.3 Pose des systèmes à résoudre

Les points que l'on cherche se divisent en 2 catégories :

- ceux dont les plans tangents sont verticaux ;
- ceux dont les plans tangents sont horizontaux.

Cela donne donc lieu à la résolution de deux systèmes :

$$\text{Points à plans verticaux} \quad \left\{ \begin{array}{l} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}^T M^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 1 \\ (x, y, z) \cdot \vec{n} = 0 \\ (x - x_0)C + (y - y_0)E + (z - z_0)F = 0 \end{array} \right. \quad (\text{B.10})$$

$$\text{Points à plans horizontaux} \quad \left\{ \begin{array}{l} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}^T M^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 1 \\ (x, \ y, \ z) \cdot \vec{n} = 0 \\ (x - x_0)B + (y - y_0)D + (z - z_0)E = 0 \end{array} \right. \quad (\text{B.11})$$

$$\text{Avec } \vec{n} = \begin{pmatrix} (x - x_0)A + (y - y_0)B + (z - z_0)C \\ (x - x_0)B + (y - y_0)D + (z - z_0)E \\ (x - x_0)C + (y - y_0)E + (z - z_0)F \end{pmatrix}$$

B.4 Résolution des systèmes

B.4.1 Résolution pour les points à plans verticaux

On rappelle que le système à résoudre est le suivant :

$$\left\{ \begin{array}{l} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix}^T M^{-1} \begin{pmatrix} x - x_0 \\ y - y_0 \\ z - z_0 \end{pmatrix} = 1 \\ (x, \ y, \ z) \cdot \vec{n} = 0 \\ (x - x_0)C + (y - y_0)E + (z - z_0)F = 0 \end{array} \right. \quad \begin{array}{l} (\text{B.12a}) \\ (\text{B.12b}) \\ (\text{B.12c}) \end{array}$$

$$\text{Avec } \vec{n} = \begin{pmatrix} (x - x_0)A + (y - y_0)B + (z - z_0)C \\ (x - x_0)B + (y - y_0)D + (z - z_0)E \\ (x - x_0)C + (y - y_0)E + (z - z_0)F \end{pmatrix}$$

$$\text{Et } M^{-1} = \begin{pmatrix} A & B & C \\ B & D & E \\ C & E & F \end{pmatrix}$$

De la relation [B.12c](#), on peut extraire que :

$$(x - x_0)C + (y - y_0)E + (z - z_0)F = 0 \Leftrightarrow z = z_0 - \frac{(x - x_0)C + (y - y_0)E}{F} \quad (\text{B.13})$$

En reprenant les équations [B.12a](#) et [B.12b](#), et en introduisant la contrainte [B.13](#), on arrive finalement au système suivant :

$$\left\{ \begin{array}{l} \begin{pmatrix} x \\ y \end{pmatrix}^T K \begin{pmatrix} x \\ y \end{pmatrix} - 2 \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}^T K \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}^T K \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = 1 \\ \begin{pmatrix} x \\ y \end{pmatrix}^T K \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}^T K \begin{pmatrix} x \\ y \end{pmatrix} = 0 \end{array} \right. \quad \begin{array}{l} \text{(B.14a)} \\ \text{(B.14b)} \end{array}$$

$$\text{Avec } K = \begin{pmatrix} A - \frac{C^2}{F} & B - \frac{CE}{F} \\ B - \frac{CE}{F} & D - \frac{E^2}{F} \end{pmatrix}$$

En remplaçant à présent B.14b dans B.14a, on aboutit à la relation liant directement x à y :

$$y = -\frac{p_0}{p_1}x - \frac{r}{p_1} \quad \text{(B.15)}$$

$$\text{Avec } \left\{ \begin{array}{l} p_0 = -\begin{pmatrix} x_0 & y_0 \end{pmatrix} \begin{pmatrix} A - \frac{C^2}{F} \\ B - \frac{CE}{F} \end{pmatrix} \\ p_1 = -\begin{pmatrix} x_0 & y_0 \end{pmatrix} \begin{pmatrix} B - \frac{CE}{F} \\ D - \frac{E^2}{F} \end{pmatrix} \\ r = \begin{pmatrix} x_0 & y_0 \end{pmatrix} K \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} - 1 \end{array} \right.$$

Enfin, en intégrant B.15 dans B.14a, on résout un polynôme d'ordre 2 en x :

$$\alpha x^2 + \beta x + \gamma = 0 \quad \text{(B.16)}$$

$$\text{Avec } \left\{ \begin{array}{l} \alpha = \begin{pmatrix} 1 \\ -\frac{p_0}{p_1} \end{pmatrix}^T K \begin{pmatrix} 1 \\ -\frac{p_0}{p_1} \end{pmatrix} \\ \beta = 2 \times \left[\left(D - \frac{E^2}{F} \right) \frac{p_0 r}{p_1^2} - \left(B - \frac{EC}{F} \right) \frac{r}{p_1} \right] \\ \gamma = \left(D - \frac{E^2}{F} \right) \left(\frac{r}{p_1} \right)^2 - r \end{array} \right.$$

Finalement, on trouve bien deux x solutions de B.16 ce qui est cohérent avec le nombre de tangentes verticales à une ellipse. Les relations B.15 et B.13 permettent de trouver les coordonnées y et z correspondantes :

$$\left\{ \begin{array}{l} \Delta_v = \beta^2 - 4\alpha\gamma \\ x_{v1} = \frac{-\beta + \sqrt{\Delta_v}}{2\alpha} \quad \left| \quad x_{v2} = \frac{-\beta - \sqrt{\Delta_v}}{2\alpha} \right. \\ y_{v1} = -\frac{p_0}{p_1}x_{v1} - \frac{r}{p_1} \quad \left| \quad y_{v2} = -\frac{p_0}{p_1}x_{v2} - \frac{r}{p_1} \right. \\ z_{v1} = z_0 - \frac{(x_{v1}-x_0)C+(y_{v1}-y_0)E}{F} \quad \left| \quad z_{v2} = z_0 - \frac{(x_{v2}-x_0)C+(y_{v2}-y_0)E}{F} \right. \end{array} \right.$$

B.4.2 Résolution pour les points à plan horizontaux

En résolvant le système B.11 de la même manière que pour le système B.10, on trouve alors :

$$\left\{ \begin{array}{l} K' = \begin{pmatrix} A - \frac{B^2}{D} & C - \frac{BE}{D} \\ C - \frac{BE}{D} & F - \frac{E^2}{D} \end{pmatrix} \\ p'_0 = - (x_0 \ y_0) \begin{pmatrix} A - \frac{B^2}{D} \\ C - \frac{BE}{D} \end{pmatrix} \quad \left| \quad p'_1 = - (x_0 \ y_0) \begin{pmatrix} C - \frac{BE}{D} \\ F - \frac{E^2}{D} \end{pmatrix} \\ r' = (x_0 \ y_0) K' \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} - 1 \\ \alpha' = \begin{pmatrix} 1 \\ -\frac{p'_0}{p'_1} \end{pmatrix}^T K' \begin{pmatrix} 1 \\ -\frac{p'_0}{p'_1} \end{pmatrix} \quad \left| \quad \gamma' = \left(F - \frac{E^2}{D} \right) \left(\frac{r'}{p'_1} \right)^2 - r' \\ \beta' = 2 \times \left[\left(F - \frac{E^2}{D} \right) \frac{p'_0 r'}{p'^2_1} - \left(C - \frac{BE}{D} \right) \frac{r'}{p'_1} \right] \\ \Delta_h = \beta'^2 - 4\alpha'\gamma' \\ x_{h1} = \frac{-\beta' + \sqrt{\Delta_h}}{2\alpha'} \quad \left| \quad x_{h2} = \frac{-\beta' - \sqrt{\Delta_h}}{2\alpha'} \\ z_{h1} = -\frac{p'_0}{p'_1} x_{h1} - \frac{r'}{p'_1} \quad \left| \quad z_{h2} = -\frac{p'_0}{p'_1} x_{h2} - \frac{r'}{p'_1} \\ y_{h1} = y_0 - \frac{(x_{h1} - x_0)B + (z_{h1} - z_0)E}{D} \quad \left| \quad y_{h2} = y_0 - \frac{(x_{h2} - x_0)B + (z_{h2} - z_0)E}{D} \end{array} \right. \right.$$

B.5 Conclusion

En conclusion, on est capable de trouver, dans le plan image, une boîte englobant la projection de l'ellipsoïde correspondant à l'incertitude associée à un point 3D. De plus, en remarquant que les différentes variables sont à chaque fois fonction de la même partie de la matrice K ou K' , on peut simplifier les écritures. Ainsi, en choisissant tour à tour $H = K$ ou $H = K'$ et en notant H_i la i^{me} colonne de H , les coordonnées ont exactement les mêmes équations :

$$\left\{ \begin{array}{l} H = \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} \\ p_0 = -(x_0 \ y_0) H_1 \quad \Bigg| \quad p_1 = -(x_0 \ y_0) H_2 \\ r = (x_0 \ y_0) H \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} - 1 \\ \alpha = \begin{pmatrix} 1 & -\frac{p_0}{p_1} \end{pmatrix} H \begin{pmatrix} 1 \\ -\frac{p_0}{p_1} \end{pmatrix} \quad \Bigg| \quad \gamma = h_{22} \left(\frac{r}{p_1} \right)^2 - r \\ \beta = -2 \frac{r}{p_1} \begin{pmatrix} 1 & -\frac{p_0}{p_1} \end{pmatrix} H_2 \\ \Delta = \beta^2 - 4\alpha\gamma \\ x_{vi} = \frac{-\beta + \sqrt{\Delta}}{2\alpha} \quad \Bigg| \quad x_{hi} = \frac{-\beta - \sqrt{\Delta}}{2\alpha} \\ y_{vi} = -\frac{p_0}{p_1} x_{vi} - \frac{r}{p_1} \quad \Bigg| \quad z_{hi} = -\frac{p_0}{p_1} x_{hi} - \frac{r}{p_1} \\ z_{vi} = z_0 - \frac{(x_{vi} - x_0)C + (y_{vi} - y_0)E}{F} \quad \Bigg| \quad y_{hi} = y_0 - \frac{(x_{hi} - x_0)B + (z_{hi} - z_0)E}{D} \end{array} \right.$$

Extraction d'une ellipse 2D à partir de deux paires de points aux tangentes parallèles deux à deux

C.1 Introduction

Nous allons montrer qu'il est possible de remonter à la forme quadratique d'une ellipse d'incertitude en connaissant 4 points de celle-ci dont les tangentes sont deux à deux égales et dans notre cas, orthogonales.

C.2 Problématique

Comme le montre la figure C.1, on connaît les 4 points d'une ellipse qui forment deux à deux les tangentes horizontales et verticales. La question est donc la suivante : avec uniquement la connaissance de ces 4 points et des valeurs des tangentes en ces points, comment retrouver la matrice de la forme quadratique associée à l'ellipse ?

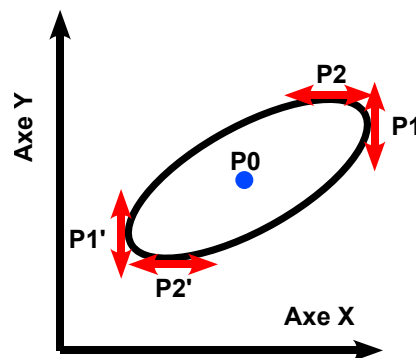


FIGURE C.1 : Schématisation du problème.

C.3 Calcul et résolution

Dans un premier temps, on s'appliquera à écrire le ou les systèmes solutions et dans un deuxième temps, on viendra les analyser pour trouver une solution.

C.3.1 Systèmes solutions

On constatera aisément que les points sont deux à deux liés par une symétrie centrale de centre le centre de l'ellipse (l'intersection des deux segments, chacun formés par les deux points aux tangentes égales, n'est autre que le centre de l'ellipse). Par conséquent, il n'y a que 2 équations indépendantes.

De plus, si une matrice est symétrique et inversible, alors son inverse est aussi symétrique, ce qui fait passer le nombre de paramètres de 4 à 3.

Comme dernière équation, il faut se servir de l'information de la tangente. En effet, si on note \vec{n} la normale à une courbe f en (x_t, y_t) on a la relation suivante :

$$\overrightarrow{\text{grad}}(f(x_t, y_t)) = \vec{n} \quad (\text{C.1})$$

On sait que les tangentes aux 4 points qui nous concernent sont soit verticales, soit horizontales ce qui engendre des normales respectivement horizontales ou verticales. Si un point se trouve sur une ellipse, alors sa distance de Mahalanobis avec le centre de l'ellipse est égale à 1 :

$$\begin{pmatrix} x_i - x_0 \\ y_i - y_0 \end{pmatrix}^T M^{-1} \begin{pmatrix} x_i - x_0 \\ y_i - y_0 \end{pmatrix} = 1 \quad (\text{C.2})$$

Avec $M = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$. Cette équation développée pour chacun des deux points (on prendra évidemment des points dont les tangentes sont différentes) donne la première partie du système :

$$\begin{cases} a(x_1 - x_0)^2 & 2b(x_1 - x_0)(y_1 - y_0) & c(y_1 - y_0)^2 = 1 \\ a(x_2 - x_0)^2 & 2b(x_2 - x_0)(y_2 - y_0) & c(y_2 - y_0)^2 = 1 \end{cases} \quad (\text{C.3})$$

Il manque la dernière relation et pour la suite de la démonstration, posons que le point (x_1, y_1) est à tangente verticale tandis que le point (x_2, y_2) est à tangente horizontale.

Le gradient de la fonction de Mahalanobis donne le résultat suivant :

$$\overrightarrow{\text{grad}}(f) = \begin{pmatrix} a(x_i - x_0) + b(y_i - y_0) \\ b(x_i - x_0) + c(y_i - y_0) \end{pmatrix} \quad (\text{C.4})$$

Pour les deux points choisis, selon l'orientation de la tangente, une des composantes du gradient est nulle ce qui nous donne la troisième équation de nos systèmes :

$$(x_1, y_1) \Rightarrow \left\{ \begin{pmatrix} (x_1 - x_0)^2 & 2(x_1 - x_0)(y_1 - y_0) & (y_1 - y_0)^2 \\ (x_2 - x_0)^2 & b(x_2 - x_0)(y_2 - y_0) & (y_2 - y_0)^2 \\ 0 & (x_1 - x_0) & (y_1 - y_0) \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \right. \quad (\text{C.5})$$

$$(x_2, y_2) \Rightarrow \left\{ \begin{pmatrix} (x_1 - x_0)^2 & 2(x_1 - x_0)(y_1 - y_0) & (y_1 - y_0)^2 \\ (x_2 - x_0)^2 & b(x_2 - x_0)(y_2 - y_0) & (y_2 - y_0)^2 \\ (x_2 - x_0) & (y_2 - y_0) & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \right. \quad (\text{C.6})$$

Pour plus de commodité, la matrices systèmes seront notées φ_1 et φ_2 .

Les systèmes sont posés, voyons dans un premier temps s'ils sont solvables. Pour cela, analysons les déterminants :

$$\begin{aligned}\Delta &= [(x_2 - x_0)(y_1 - y_0) - (x_1 - x_0)(y_2 - y_0)]^2 \\ \det(\varphi_1) &= -(x_1 - x_0)\Delta\end{aligned}\tag{C.7}$$

$$\det(\varphi_2) = -(y_2 - y_0)\Delta\tag{C.8}$$

Par conséquent, on peut émettre la résolution conditionnelle suivante :

- $\mathbf{y}_1 = \mathbf{y}_2 = \mathbf{y}_0$ alors $x_1 = x_2 = x_0$ et par conséquent l'ellipse est donc parallèle aux axes et on en déduit la matrice diagonale de la forme quadratique ;
- $\mathbf{x}_2 = \mathbf{x}_1 = \mathbf{x}_0$ alors $y_1 = y_2 = y_0$ et par conséquent l'ellipse est donc parallèle aux axes et on en déduit la matrice diagonale de la forme quadratique ;
- le dernier cas est tout simplement celui d'une orientation de l'ellipse quelconque qui pose un système isostatique.

Finalement, on s'aperçoit que les deux systèmes sont équivalents. On se propose alors de n'étudier que le système φ_2 . Une résolution par moindres carrés est réalisable en première intention et en voici la forme :

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = [(\varphi_2^T \varphi_2)^{-1} \varphi_2^T] \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}\tag{C.9}$$

$$M = \begin{pmatrix} a & b \\ b & c \end{pmatrix}^{-1} = \frac{1}{ac - b^2} \begin{pmatrix} c & -b \\ -b & a \end{pmatrix}\tag{C.10}$$

Une deuxième méthode serait de développer les calculs pour obtenir directement les valeurs (a, b, c) sans passer par la pseudo-inverse. On trouve alors les relations suivantes :

$$\begin{aligned}\Delta &= [(x_2 - x_0)(y_1 - y_0) - (x_1 - x_0)(y_2 - y_0)]^2 \\ \det(\varphi_2) &= -(y_2 - y_0)\Delta \\ a &= [(y_2 - y_0)(y_1 - y_0)^2 - (y_2 - y_0)^3]\end{aligned}\tag{C.11}$$

$$b = [(x_2 - x_0)((y_2 - y_0)^2 - (y_1 - y_0)^2)]\tag{C.12}$$

$$\begin{aligned}c &= [2(x_1 - x_0)(x_2 - x_0)(y_1 - y_0) - \\ &\quad (x_1 - x_0)^2(y_2 - y_0) - (x_2 - x_0)^2(y_2 - y_0)]\end{aligned}\tag{C.13}$$

$$M = \frac{1}{\det(\varphi_2)} \begin{pmatrix} a & b \\ b & c \end{pmatrix}^{-1} = \frac{1}{(ac - b^2)\det(\varphi_2)} \begin{pmatrix} c & -b \\ -b & a \end{pmatrix}\tag{C.14}$$

C.4 Conclusion

En conclusion, nous venons de voir qu'il est possible de reconstruire intégralement la forme quadratique associée à une ellipse d'incertitude dès lors que l'on connaît 4 points et dont les tangentes sont deux à deux égales. Le cas des tangentes orthogonales permet de simplifier les calculs.

Estimation de la profondeur d'initialisation des points 3D par une approche « monde plan »

D.1 Introduction

L'observation d'un point par une caméra ne permet pas de reconstruire un point 3D. Ce capteur ne permet pas une estimation de la profondeur et ce n'est qu'avec une seconde observation du même point depuis un point de vue que le problème peut se résoudre (sous réserve d'une parallaxe suffisante entre les deux positions d'observation). Cependant, il est tout de même possible de construire un point 3D à partir d'une seule observation en définissant une profondeur par défaut. Pour pouvoir prendre en compte les points éloignés comme les points proches, il convient de mettre cette incertitude quasiment à la limite d'observation, fournissant un point de fonction parfois incohérent avec la réalité. Nous allons voir que, dans une approche « monde plan », nous pouvons définir une profondeur d'initialisation plus pertinente.

D.2 Problématique

On se donne un point dans l'image dont les coordonnées sont (u, v) . Cette image est obtenue par l'observation du monde dont les paramètres intrinsèques sont définis par la matrice \mathbf{K} :

$$\mathbf{K} = \begin{pmatrix} 1 & 0 & 0 \\ C_u & 0 & f_u \\ C_v & f_v & 0 \end{pmatrix} \quad (\text{D.1})$$

où (f_u, f_v) sont respectivement les focales sur les axes \vec{U} et \vec{V} et où (C_u, C_v) définissent les coordonnées du point centrale dans l'image. De plus, la caméra est définie, dans le repère du robot par 3 paramètres de translation (t_x, t_y, t_z) et par 3 angles de rotation $(\alpha_x, \alpha_y, \alpha_z)$. La figure D.1 illustre la position du capteur dans le repère robot et fournit les variables de calcul. On se propose alors de trouver, sous la contrainte d'un monde plan, une meilleure estimation de la profondeur X_0 .

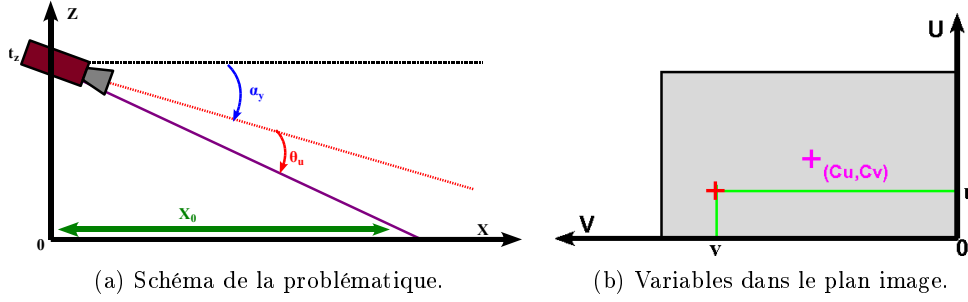


FIGURE D.1 : Schéma du problème de calcul de la profondeur d'initialisation. On rappelle que

D.3 Résolution

L'illustration D.1a permet d'écrire la relation liant X_0 , θ_u et α_y :

$$X_0 = \frac{t_z}{\tan(\alpha_y + \theta_u)} \quad (D.2)$$

L'expression de θ_u se trouve à l'aide des variables de l'illustration D.1b :

$$\theta_u = \arctan\left(\frac{C_u - u}{f_u}\right) \quad (D.3)$$

Constatons que, faute de connaissance de l'environnement, et sous l'hypothèse d'un monde plan, il ne faut considérer ce calcul que lorsque $C_u - u > 0$ ($C_u - u < 0$ engendrerait un $X_0 < 0$). De plus, la condition limite $u \rightarrow C_u$ ferait tendre le calcul de X_0 vers l'infini. Par conséquent, il convient de fixer, à présent, une contrainte limite X_{max} de telle sorte que :

$$X_0 = \begin{cases} X_{max} & \text{si } C_u - u > 0 \text{ ou } X_0 > X_{max} \\ X_0 & \text{sinon} \end{cases} \quad (D.4)$$

D.4 Conclusion

Nous avons vu comment créer, sous l'hypothèse d'un monde plan, la profondeur d'initialisation des points 3D. De plus, nous avons donné les conditions de définition permettant, non plus d'avoir une profondeur fixe pour chaque point, mais une profondeur variable entre l'observateur et une valeur maximale fixée.

Construction contrôlée d'ellipsoïde

E.1 Méthode des jacobienues

E.1.1 Introduction

On se donne un point 3D du monde que l'on vient de créer à partir de son observation dans une image :

$$\begin{pmatrix} u \\ v \end{pmatrix} \xrightarrow{X_0} \begin{pmatrix} y \\ z \end{pmatrix} \rightarrow \vec{P}_{t_{3D}} = \begin{pmatrix} X_0 \\ y \\ z \end{pmatrix} \quad (\text{E.1})$$

A partir de ce point et en ayant fixé σ_u et σ_v au préalable, est-il possible de trouver une bonne estimation de la matrice de variance-covariance associée à ce point de telle sorte que la projection de l'incertitude par les jacobienues soit cohérente avec l'incertitude fixée dans l'image ?

E.1.2 Problématique

Si on pose un point $\vec{P}_{t_{3D}}$ de l'espace 3D avec $\mathbf{P}_{P_{t_{3D}}}$ la matrice de covariance associée et $\mathbf{g}(\vec{P}_{t_{3D}})$ une fonction d'observation de $\vec{P}_{t_{3D}}$ non-linéaire, alors la propagation de l'incertitude $\mathbf{P}_{P_{t_{3D}}}$ sera estimée au premier ordre dans l'espace image de \mathbf{g} par :

$$\mathbf{P}_{2D} = \mathbf{G} \mathbf{P}_{P_{t_{3D}}} \mathbf{G}^T \quad \text{avec} \quad \mathbf{G} = \left. \frac{\partial \mathbf{g}}{\partial \vec{P}_{t_{3D}}} \right|_{\vec{P}_{t_{3D}}} \quad (\text{E.2})$$

La question est de savoir si, connaissant \mathbf{P}_{2D} , il est possible de remonter aux coefficients de $\mathbf{P}_{P_{t_{3D}}}$? En d'autres termes, quelle incertitude 3D a pour projection \vec{P}_{2D} par \mathbf{g}

E.1.3 Résolution

E.1.3.1 Variables du contexte

La matrice de projection \mathbf{K} est définie de la manière suivante :

$$\mathbf{K} = \begin{pmatrix} K_1 \\ K_2 \\ K_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ C_u & 0 & f_u \\ C_v & f_v & 0 \end{pmatrix} \quad (\text{E.3})$$

$$\vec{P}_{t_{2D}} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{K_2 \vec{P}_{t_{3D}}}{K_1 \vec{P}_{t_{3D}}} \\ \frac{K_3 \vec{P}_{t_{3D}}}{K_1 \vec{P}_{t_{3D}}} \end{pmatrix} \quad (\text{E.4})$$

On se donne le point $\vec{P}t_{3D}$ ainsi que la matrice d'incertitude définie sur l'axe principale $\mathbf{P}_{Pt_{3D}}$ et son projeté 2D $\vec{P}t_{2D}$.

E.1.3.2 Résolution littérale

On calcule littéralement la jacobienne sur u et v :

$$\begin{array}{l|l} u = \frac{C_u x + f_u z}{x} & v = \frac{C_v x + f_v y}{x} \\ \frac{\partial u}{\partial x} = \frac{-f_u z}{x^2} & \frac{\partial v}{\partial x} = \frac{-f_v y}{x^2} \\ \frac{\partial u}{\partial y} = 0 & \frac{\partial v}{\partial y} = \frac{f_v}{x} \\ \frac{\partial u}{\partial z} = \frac{f_u}{x} & \frac{\partial v}{\partial z} = 0 \end{array}$$

Par conséquent \mathbf{G} s'écrit :

$$\mathbf{G} = \begin{pmatrix} \partial u / \partial x & \partial u / \partial y & \partial u / \partial z \\ \partial v / \partial x & \partial v / \partial y & \partial v / \partial z \end{pmatrix} = \begin{pmatrix} -f_u z / x^2 & 0 & f_u / x \\ -f_v y / x^2 & f_v / x & 0 \end{pmatrix} \quad (\text{E.5})$$

Ainsi, on peut calculer à présent la propagation d'incertitude de \mathbf{P}_{Pt} par \mathbf{g} :

$$\mathbf{P}_{2D} = \mathbf{G} \mathbf{P}_{Pt} \mathbf{G}^T = \begin{pmatrix} \sigma_x^2 \frac{f_u^2 z^2}{x^4} + \sigma_z^2 \frac{f_u^2}{x^2} & \sigma_x^2 \frac{f_u f_v y z}{x^4} \\ \sigma_x^2 \frac{f_u f_v y z}{x^4} & \sigma_x^2 \frac{f_v^2 y^2}{x^4} + \sigma_y^2 \frac{f_v^2}{x^2} \end{pmatrix} \quad (\text{E.6})$$

E.1.4 Cas particulier

Le cas d'un point sur l'axe principal est un cas particulier qui permet de simplifier grandement les relations. En effet, nous pouvons déplacer l'incertitude où nous le souhaitons et l'orienter comme nous le souhaitons au travers des matrices de rotation. Ainsi, en prenant le point $\vec{P}t_0$ sur l'axe principal (l'axe de \vec{x}) et en développant \mathbf{P}_{2D} qui se simplifie dans ce cas :

$$\begin{aligned} \rho &= \sqrt{x^2 + y^2 + z^2} \\ \vec{P}t_0 &= \begin{pmatrix} \rho \\ 0 \\ 0 \end{pmatrix} \end{aligned} \quad (\text{E.7})$$

$$\mathbf{P}_{2D} = \begin{pmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_v^2 \end{pmatrix} = \begin{pmatrix} \sigma_z^2 \frac{f_u^2}{\rho^2} & 0 \\ 0 & \sigma_y^2 \frac{f_v^2}{\rho^2} \end{pmatrix} \quad (\text{E.8})$$

E.1.5 Conclusion

Par une simple identification à partir de l'égalité E.8, on trouve les relations de définition de σ_y et σ_z :

$$\sigma_x = \rho - \delta \quad (\text{E.9})$$

$$\sigma_y = \rho \frac{\sigma_v}{f_v} \quad (\text{E.10})$$

$$\sigma_z = \rho \frac{\sigma_u}{f_u} \quad (\text{E.11})$$

$$(\text{E.12})$$

avec δ qui définit la distance minimale d'observation afin de garantir que l'origine du repère n'est pas incluse dans l'ellipsoïde d'incertitude généré. Nous avons donc pu calculer un ellipsoïde d'incertitude connaissant une incertitude 2D définie dans une image et compte tenu de la fonction d'observation. Cette méthode garantie que la projection de l'incertitude est conforme à l'incertitude d'observation.

E.2 Méthode de la projection d'ellipsoïde

E.2.1 Introduction

On se donne un point 3D du monde que l'on vient de créer à partir de son observation dans une image :

$$\begin{pmatrix} u \\ v \end{pmatrix} \xrightarrow{X_0} \begin{pmatrix} y \\ z \end{pmatrix} \rightarrow \vec{P}_{t3D} = \begin{pmatrix} X_0 \\ y \\ z \end{pmatrix} \quad (\text{E.13})$$

A partir de ce point et en ayant fixé σ_u et σ_v au préalable, est-il possible de trouver une bonne estimation de la matrice de variance-covariance associée à ce point de telle sorte que la projection des points situés à 1σ de l'estimation soit cohérente avec l'incertitude fixée dans l'image.

E.2.2 Problématique

Ce problème se pose de manière identique sur \vec{y} et sur \vec{z} . Nous allons donc l'aborder sous sa forme bi-dimensionnelle comme l'illustre la figure E.1 Nous savons

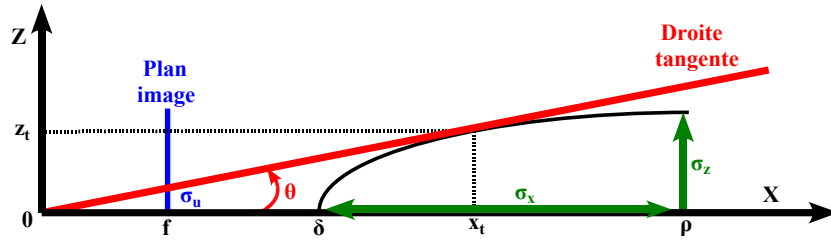


FIGURE E.1 : Schéma de la problématique

construire une ellipse et calculer la matrice de rotation de telle sorte qu'une fois déplacée, sa projection soit cohérente avec une incertitude dans l'image. Si on définit l'ellipse ainsi :

$$\frac{(x - x_0)^2}{a^2} + \frac{(z - z_0)^2}{b^2} = 1 \quad (\text{E.14})$$

avec (x_0, y_0) le centre de l'ellipse et (a, b) les longueurs des axes, alors l'équation de la tangente au point (x_t, y_t) est de la forme :

$$\frac{(x - x_0)(x_t - x_0)}{a^2} + \frac{(z - z_0)(z_t - z_0)}{b^2} = 1 \quad (\text{E.15})$$

E.2.3 Résolution

D'après le schéma de notre problématique, on peut écrire les identifications suivantes :

$$\begin{aligned} a &= \sigma_x = \sqrt{X_0^2 + y^2 + z^2} - \delta = \rho - \delta \\ b &= \sigma_z \\ x_0 &= X_0 \\ z_0 &= 0 \end{aligned} \tag{E.16}$$

où δ définit la distance minimale d'observation, pour que l'ellipse ne contienne pas l'origine du repère. En remplaçant dans la relation de la tangente, on trouve :

$$\frac{(x - \rho)(x_t - \rho)}{(\rho - \delta)2} + \frac{zz_t}{\sigma_z^2} = 1 \tag{E.17}$$

De plus, z_t est défini tel que :

$$\tan(\theta) = \frac{z_t}{x_t} = \frac{\sigma_u}{f} \tag{E.18}$$

$$\Leftrightarrow z_t = x_t \frac{\sigma_u}{f} \tag{E.19}$$

d'où

$$\frac{(x - \rho)(x_t - \rho)}{(\rho - \delta)2} + xx_t \frac{\sigma_u^2}{f^2 \sigma_z^2} = 1 \tag{E.20}$$

Il faut à présent trouver deux points qui vérifient la relation pour pouvoir déterminer dans ce cas σ_z . On choisit les points définis à $x = 0$ et $x = \rho$. On obtient le système suivant :

$$\begin{cases} \rho x_t \frac{\sigma_u^2}{f^2 \sigma_z^2} = 1 \\ -\rho \frac{(x_t - \rho)}{(\rho - \delta)^2} = 1 \end{cases} \tag{E.21}$$

Finalement, en combinant ces deux relations on trouve la relation cherchée :

$$\sigma_z = \frac{\sigma_u}{f} \sqrt{\rho^2 - \sigma_x^2} \tag{E.22}$$

E.2.4 Conclusion

On trouve que, pour un couple (σ_u, σ_v) donné dans l'image, il est possible d'exprimer le couple (σ_y, σ_z) de façon simple en fonction d'un σ_x donné :

$$\begin{aligned} \rho &= \sqrt{X_0^2 + y^2 + z^2} \\ \sigma_x &= \rho - \delta \end{aligned} \tag{E.23}$$

$$\sigma_y = \frac{\sigma_v}{f} \sqrt{\rho^2 - \sigma_x^2} \tag{E.24}$$

$$\sigma_z = \frac{\sigma_u}{f} \sqrt{\rho^2 - \sigma_x^2} \tag{E.25}$$

$$\tag{E.26}$$

On vérifie d'ailleurs que la condition d'existence de ces relations est $\rho > \sigma_x$, ce qui traduit que l'ellipse ne doit pas englober l'origine du repère.

Bibliographie

- [Agrawal 2006] M. Agrawal et K. Konolige. *Real-time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS*. 18th International Conference on Pattern Recognition, 2006. (Cité en page 13)
- [Bar-Shalom 1988] Y. Bar-Shalom et T. E. Fortmann. Tracking and data association. 1988. (Cité en page 25)
- [Bevly 2004] D. M. Bevly. *Global Positioning System (GPS) : A Low-Cost Velocity Sensor for Correcting Inertial Sensor Errors on Ground Vehicles*. Journal of Dynamic Systems, Measurement, and Control, vol. 126, 2004. (Cité en page 13)
- [Bresson 2011] G. Bresson, T. Féraud, R. Aufrère, P. Checchin et R. Chapuis. *A New Strategy for Feature Initialization in Visual SLAM*. In IROS11 Workshop on Perception and Navigation for Autonomous Vehicles in Human Environment, San Francisco, CALIFORNIA, September, 30th 2011. (Cité en page 56)
- [Broggi 2001] A. Broggi, M. Bertozzi et A. Fascioli. *Self-Calibration of a Stereo Vision System for Automotive Applications*. IEEE International Conference on Robotics and Automation, 2001. (Cité en page 15)
- [Bujnak 2008] M. Bujnak, Z. Kukelova et T. Pajdla. *A general solution to the p4p problem for camera with unknown focal length*. Anchorage, Alaska, USA, June 2008. 21th IEEE Conference on Computer Vision and Pattern Recognition. (Cité en page 21)
- [Campion 1996] G. Campion, G. Bastin et B. D'Andréa-Novel. *Structural properties and classification of kinematic and dynamic models of wheeled mobile robots*. IEEE Transactions on Robotics and Automation, vol. 12, pages 47–62, 1996. (Cité en page 12)
- [Castellanos 2004] J. Castellanos, J. Neira et J. Tardos J. *Limits to the consistency of EKF-based SLAM*. In 5th IFAC Symposium on Intelligent Autonomous Vehicles, 2004. (Cité en page 25)
- [Chanier 2010] F. Chanier. *Localisation et cartographie simultanées de l'environnement à bord de véhicules autonomes : analyse de solutions fondées sur le filtrage de Kalman*. PhD thesis, Université Blaise Pascal - Clermont II, Ecole Doctorale Sciences Pour l'Ingénieur De Clermont-Ferrand, 2010. (Cité en page 70)
- [Chen 2009] Xianzhong Chen. *An Adaptive UKF-Based Particle Filter for Mobile Robot SLAM*. International Joint Conference on Artificial Intelligence, pages 167–170, 2009. (Cité en page 26)
- [Civera 2009] J. Civera, O. G. Grasa, A. J. Davison et J.M.M. Montiel. *1-Point RANSAC for EKF-Based Structure from Motion*. In Proceedings IEEE/RSJ

- Int. Conf. on Intelligent Robots and Systems (IROS), St. Louis, October 2009. (Cité en pages 28 et 42)
- [Cole 2006] D. M. Cole et P. M. Newman. *Using laser range data for 3D SLAM in outdoor environments*. IEEE International Conference on Robotics and Automation, May, 15-19 2006. (Cité en page 29)
- [Csorba 1998] M. Csorba. *Simultaneous localisation and map building*. PhD thesis, University of Oxford, 1998. (Cité en page 29)
- [Davison 1998] A.J. Davison et D.W. Murray. *Mobile Robot Localisation Using Active Vision*. pages 809–825. Proc. Fifth European Conf. Computer Vision, 1998. (Cité en page 30)
- [Davison 2003] A. Davison. *Real-time simultaneous localisation and mapping with a single camera*. In Proc. International Conference on Computer Vision (ICCV), Nice, October 2003. (Cité en page 30)
- [Davison 2007] A. Davison, I. Reid, N. Molton et O. Stasse. *MonoSLAM : Real-Time Single Camera SLAM*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 6, pages 1052–1067, 2007. (Cité en page 29)
- [Delmas 2011] P. Delmas. *Génération active des déplacements d'un véhicule agricole dans son environnement*. PhD thesis, Université Blaise Pascal - Clermont II, 2011. (Cité en pages 39 et 50)
- [Féraud 2010a] T. Féraud, P. Checchin, R. Aufrère et R. Chapuis. *Communicating Vehicles in Convoy and Monocular Vision-based Localization*. Lecce - ITALY, September, 7-9 2010. 7th IFAC Symposium on Intelligent Autonomous Vehicles. (Cité en page 30)
- [Féraud 2010b] T. Féraud, P. Checchin et R. Chapuis. *Véhicules communicants en convoi et localisation par vision monoculaire*. Caen - FRANCE, January, 19-22 2010. 17ème congrès francophone AFRIF-AFIA, Reconnaissance des Formes et Intelligence Artificielle. RFIA 2010. (Cité en page 47)
- [Féraud 2011a] T. Féraud, R. Chapuis, R. Aufrère et P. Checchin. *Improving Results of Rational Non-Linear Observation Functions Using a Kalman Filter Correction*. Chicago, ILLINOIS, July, 4-8 2011. 14th International Conference on Information Fusion. (Cité en page 89)
- [Féraud 2011b] T. Féraud, R. Chapuis, R. Aufrère et P. Checchin. *Kalman filter correction with non-linear function : Application to Visual-SLAM*. Örebro, SWEDEN, September, 7-9 2011. 5th European Conference on Mobile Robots. (Cité en page 89)
- [Fischler 1981] Martin A. Fischler et Robert C. Bolles. *Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography*. Commun. ACM, vol. 24, no. 6, pages 381–395, 1981. <http://doi.acm.org/10.1145/358669.358692>. (Cité en page 21)
- [Folkesson 2005] J. Folkesson, P. Jensfelt et H. Christensen. *Vision slam in the measurement subspace*. Barcelona, SPAIN, April 2005. ICRA-05. (Cité en page 19)

- [Forsyth 2002] D. A. Forsyth et J. Ponce. *Computer vision : A modern approach*. Prentice Hall, 2002. (Cité en page 22)
- [Garcia 2004] M. A. Garcia et A. Solanas. *3D simultaneous localization and modeling from stereo vision*. IEEE International Conference on Robotics and Automation, April 26 - May 01 2004. (Cité en page 15)
- [Garulli 2005] A. Garulli, A. Giannitrapani et A. Rossi. *Mobile robot SLAM for line-based environment representation*. 44th IEEE Conference on Decision and Control, December, 12-15 2005. (Cité en page 19)
- [Gibbens 2000] P. Gibbens, G. Dissanayake et H. Durrant-White. *A closed form solution to the single degree of freedom simultaneous localisation and map building (slam) problem*. volume 1. 39th IEEE Conference on Decision and Control, 2000. (Cité en page 26)
- [Haralick 1979] R. M. Haralick. *Statistical and structural approaches to texture*. In *Proceedings of the IEEE*, volume 67, pages 786–804, 1979. (Cité en page 72)
- [Haralick 1994] R. M. Haralick, C. N. Lee, K. Ottenberg et M. Nolle. *Review and analysis of solutions of the three point perspective pose estimation problem*. *International Journal of Computer Vision*, vol. 13, pages 331–356, 1994. (Cité en page 20)
- [Harris 1988] C. Harris et M. Stephens. *A combined corner and edge detector*. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988. (Cité en page 72)
- [Hartley 1995] R. Hartley. *In defence of the 8-point algorithm*. pages pp.1064–1070. *Fifth International Conference on Computer Vision*, 1995. (Cité en page 22)
- [Hurn 1998] J. Hurn. *Gps : A guide to the next utility*. Trimble Navigation, 1998. (Cité en page 10)
- [Jensfelt 2001] P. Jensfelt et S. Kristensen. *Active global localisation for a mobile robot using multiple hypothesis tracking*. *IEEE Transactions on Robotics and Automation*, vol. 17, pages 748–760, 2001. (Cité en page 13)
- [Joly 2010] C. Joly. *Contributions aux méthodes de localisation et cartographie simultanées par vision omnidirectionnelle*. PhD thesis, École Nationale Supérieure des Mines de Paris, 2010. (Cité en page 100)
- [Julier 2000] S. Julier, J. Uhlmann et H. F. Durrant-White. *A new method for nonlinear transformation of means and covariances in filters and estimators*. *IEEE Trans Automatic Control*, vol. vol.45, pages 477–482, March 2000. (Cité en page 26)
- [Julier 2004] J. Julier et J. K. Uhlmann. *Unscented filtering and nonlinear estimation*. *Proc. IEEE*, vol. 92, pages 401–422, 2004. (Cité en page 26)
- [Konolige 2008] K. Konolige et M. Agrawal. *Frameslam : From bundle adjustment to real-time visual mapping*. volume 24, pages 1066–1077. *IEEE Transactions on Robotics*, 2008. (Cité en page 24)

- [Lemaire 2005] T. Lemaire, S. Lacroix et J. Sola. *A practical 3d Bearing-Only slam algorithm*. pages 2449–2454. 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), August 2005. (Cité en page 19)
- [Lenain 2005] R. Lenain, B. Thuilot, C. Cariou et P. Martinet. *Model predictive control for vehicle guidance in presence of sliding : application to farm vehicles path tracking*. pages 897–902, Barcelona, Spain, April 2005. In International Conference on Robotics and Automation. (Cité en page 12)
- [Leonard 1991] J. J. Leonard et H. F. Durrant-whyte. *Simultaneous map building and localization for an autonomous mobile robot*. In Proc. IROS, in IEEE/RSJ International Workshop, Intelligence for Mechanical Systems, pages 1442–1447, 1991. (Cité en page 29)
- [Li 2006] H. Li et R. Hartley. *Five-point motion estimation made easy*. pages 630–633, Washington, DC, USA, 2006. 18th International Conference on Pattern Recognition. (Cité en page 22)
- [Longuet-Higgins 1981] H. C. Longuet-Higgins. Readings in computer vision : Issues, problems, principles, and paradigms de m. a. fischler et o. firschein, chapitre A computer algorithm for reconstructing a scene from two projections, pages 61–62. 1981. (Cité en page 22)
- [Lowe 1999] D. Lowe. *Object recognition from local scale-invariant features*. In Proceedings of the International Conference on Computer Vision, volume 2, pages 1150–1157, 1999. (Cité en page 72)
- [Lowe 2004] D. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, vol. 60, pages 91–110, 2004. (Cité en page 72)
- [Maghami 2007] M. Maghami, R. A. Zoroofi et B. N. Araabi. *Kalman filter tracking for facial expression recognition using noticeable feature selection*. International Conference on Intelligent and Advanced Systems, November, 25-28 2007. (Cité en page 25)
- [Mahon 2008] I. Mahon, S.B. Williams, O. Pizarro et M. Johnson-Roberson. *Efficient View-Based SLAM Using Visual Loop Closures*. IEEE Trans. on Robotics and Automation, vol. 24, no. 5, pages 1002–1014, October 2008. (Cité en page 26)
- [Malartre 2011] F. Malartre. *Perception intelligente pour la navigation rapide de robots mobiles en environnement naturel*. PhD thesis, Ecole Doctorale Science Pour l'Ingénieur de Clermont Ferrand, Université Blaise Pascal, 2011. (Cité en pages 15, 28, 50 et 73)
- [Martin 2008] P. Martin et S. Julier. *Avoiding Negative Depth in Inverse Depth Bearing-Only SLAM*. In IEEE/RSJ Int. Conf. on intelligent Robots an Systems, Nice - FRANCE, 2008. (Cité en pages 92 et 98)
- [Mikolajczyk 2004a] K. Mikolajczyk et C. Schmid. *Scale and Affine Invariant Interest Point Detectors*. International Journal of Computer Vision, vol. 60, no. 1, pages 63–86, 2004. (Cité en page 72)

- [Mikolajczyk 2004b] K. Mikolajczyk, K. Schmid et C. Schmid. *Scale and affine invariant interest point detectors*. International Journal of Computer Vision, vol. 60, pages 63–86, 2004. (Cité en page 72)
- [Mouragnon 2009] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser et P. Sayd. *Generic and Real-Time Structure from Motion using Local Bundle Adjustment*. Image and Vision Computing, vol. 27, pages 1178–1193, 2009. (Cité en pages 24 et 29)
- [Mourllion 2006] B. Mourllion, D. Gruyer et A. Lambert. *Variance behavior and signification in probabilistic framework applied to vehicle localization*. pages 294–299, Tokyo, JAPAN, 2006. In IEEE/ITSS International Conference on Intelligent Vehicle. (Cité en page 24)
- [Neira 2001] J. Neira et J. D. Tardós. *Data Association in Stochastic Mapping Using the Joint Compatibility Test*. volume 17. IEEE Transactions on Robotics And Automation, December 2001. (Cité en page 28)
- [Newman 2006] P. Newman, D. Cole et K. Ho. *Outdoor SLAM using visual appearance and laser ranging*. IEEE International Conference on Robotics and Automation, May, 15-19 2006. (Cité en page 16)
- [Nistér 2004] D. Nistér. *An efficient solution to the five-point relative pose problem*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, pages 756–777, 2004. (Cité en page 22)
- [Ohno 2005] K. Ohno et S. Tadokoro. *Dense 3D map building based on LRF data and color image fusion*. IEEE/RSJ International Conference on Intelligent Robots and Systems, August, 2-6 2005. (Cité en page 19)
- [Palmer 2006] T. Palmer et R. Hagedorn. Predictability of weather and climate. 2006. (Cité en pages 25 et 93)
- [Paz 2008] L. M. Paz, J. D. Tardós et J. Neira. *Divide and Conquer : EKF SLAM in $O(n)$* . volume 24. IEEE Transactions on Robotics, October 2008. (Cité en page 42)
- [Press 1992] W. Press, B. Flannery, S. Teukolsky et W. Vetterling. *Numerical Recipes in C : The Art of Scientific Computing*. 1992. (Cité en page 23)
- [Quan 1999] L. Quan et Z. Lan. *Linear n -point camera pose determination*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, pages 774–780, 1999. (Cité en page 20)
- [Raguram 2008] R. Raguram, J.M. Frahm et M. Pollefeys. *A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus*. In Proc. of the 10th European Conference on Computer Vision (ECCV), pages 500–513, Marseille, France, 2008. (Cité en page 21)
- [Rebeix 1910] Rebeix. Nouvelles annales de mathématiques 4ème série, chapitre Section plan d'un cône ou d'un cylindre à base elliptique, hyperbolique ou parabolique, pages p. 107–122. 1910. (Cité en page 52)

- [Renault 2005] S. Renault, A. Le Meur et D. Meizel. *GPS/SIG localization for management of vision referenced navigation in urban environments*. Intelligent Transportation Systems, pages 608–613, 2005. (Cité en page 13)
- [Ristic 2004] Arulampalam-S. Gordon N. Ristic B. Beyond the kalman filter : Particle filters for tracking applications. 2004. (Cité en pages 24, 50 et 89)
- [Rosten 2006] E. Rosten et T. Drummond. *Machine learning for high-speed corner detection*. In European Conference on Computer Vision, May 2006. (Cité en page 72)
- [Royer 2006] E. Royer, M. Lhuillier, M. Dhome et J.-M. Lavest. *Localisation par vision monoculaire pour la navigation autonome : précision et stabilité de la méthode*. Caen - FRANCE, January, 19-22 2006. 15ème congrès francophone AFRIF-AFIA, Reconnaissance des Formes et Intelligence Artificielle. RFIA 2010. (Cité en page 23)
- [Ryu 2004] J. Ryu et J. C. Gerdes. *Integrating Inertial Sensors With Global Positioning System (GPS) for Vehicle Dynamics Control*. Journal of Dynamic Systems, Measurement, and Control, vol. 126, page 12 pages, 2004. (Cité en page 13)
- [Se 2001] S. Se, D. Lowe et J. Little. *Vision-based mobile robot localization and mapping using scale-invariant features*. IEEE International Conference on Robotics and Automation, 2001. (Cité en page 18)
- [Shi 1994] J. Shi et C. Tomasi. *Good Features to Track*. In 9th IEEE Conference on Computer Vision and Pattern Recognition. Springer., June 1994. (Cité en page 72)
- [Solà 2005] J. Solà, A. Monin, M. Devy et T. Lemaire. *Undelayed initialization in bearing only SLAM*. In IEEE/RSJ Int. on intelligent Robots and Systems, pages 2499–2504, Edmonton - CANADA, 2005. (Cité en pages 15 et 19)
- [Solà 2007] J. Solà. *Towards visual localization, mapping and moving objects tracking by a mobile robot : a geometric and probabilistic approach*. PhD thesis, Institut National Polytechnique, Toulouse, February 2007. (Cité en page 106)
- [Solà 2008] J. Solà, A. Monin, M. Devy et T. Vidal-Calleja. *Fusing Monocular Information in Multi-Camera SLAM*. IEEE Trans. on Robotics and Automation, vol. 24, no. 5, pages 958–968, October 2008. (Cité en page 15)
- [Solà 2009] J. Solà, T. Vidal-Calleja et M. Devy. *Undelayed initialization of line segments in monocular SLAM*. In IEEE/RSJ Int. Conference on intelligent Robots and Systems, 2009. (Cité en page 19)
- [Stewenius 2006] H. Stewenius, C Engels et D. Nistér. *Recent developments on direct relative orientation*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 60, pages 284–294, 2006. (Cité en page 22)
- [Straka 2011] O. Straka, J. Duník et M. Simandl. *Gaussian Sum Unscented Kalman Filter with Adaptive Scaling Parameters*. Chicago, ILLINOIS, July, 5-8 2011. 14th International Conference on Information Fusion. (Cité en page 29)

- [Tessier 2007] C. Tessier. *Système de localisation basé sur une stratégie de perception cognitive appliqué à la navigation autonome d'un robot mobile*. PhD thesis, Université Blaise Pascal - Clermont Ferrand II, 2007. (Cité en page 13)
- [Thrun 2000] S. Thrun, D. Fox, W. Burgard et F. Dellaert. *Robust Monte Carlo Localization for Mobile Robots*. Artificial Intelligence, vol. 128, no. 1-2, pages 99–141, 2000. (Cité en page 29)
- [Vaganay 1993] J. Vaganay, M. J. Aldon et A. Fournier. *Mobile robot attitude estimation by fusion of inertial data*. In IEEE international Conference on Robotics and Automation, 1993. (Cité en page 13)
- [Vivet 2010] D. Vivet, P. Checchin et R. Chapuis. *Line-based SLAM with Slow Rotating Range Sensors : Results and Evaluations*. 2010 IEEE Int. Conf. on Control, Automation, Robotics and Vision (ICARCV 2010), December, 7-10 2010. ICARCV 10. (Cité en page 16)
- [Wan 2000] E. A. Wan et R. van der Merwe. *The unscented Kalman filter for non-linear estimation*. In Proc IEEE Symp - Adaptive Systems for Signal, pages 153–158, Lake Louise - CANADA, 2000. Proc., Comm. and Control (AS-SPCC). (Cité en page 26)
- [Xie 1997] M. Xie, J. Lee, S. Z. li et S. D. Ma. *Depth Assessment Using Qualitative Stereo Vision*. volume 2, pages 1446–1449, Beijing, October, 28-31 1997. IEEE First International Conference on Intelligent Processing System. (Cité en page 14)